Algèbre linéaire

David Loiseaux et Fanny Simões

28 Février 2022

Contenu

Contenu

Qu'est-ce que l'algèbre linéaire ?

Systèmes d'équations linéaires

On appelle système de m équations linéaires à n inconnues, un système de la forme :

$$\begin{cases} a_{11}x_1 + \cdots + a_{1j}x_j + \cdots + a_{1n}x_n = b_1 & L_1 \\ \vdots & \vdots \\ a_{i1}x_1 + \cdots + a_{ij}x_j + \cdots + a_{in}x_n = b_i & L_i \\ \vdots & \vdots \\ a_{m1}x_1 + \cdots + a_{mj}x_j + \cdots + a_{mn}x_n = b_m & L_m \end{cases} \tag{1}$$

Systèmes d'équations linéaires

On appelle système de m équations linéaires à n inconnues, un système de la forme :

$$\begin{cases} a_{11}x_1 + \cdots + a_{1j}x_j + \cdots + a_{1n}x_n = b_1 & L_1 \\ \vdots & \vdots \\ a_{i1}x_1 + \cdots + a_{ij}x_j + \cdots + a_{in}x_n = b_i & L_i \\ \vdots & \vdots \\ a_{m1}x_1 + \cdots + a_{mj}x_j + \cdots + a_{mn}x_n = b_m & L_m \end{cases} \tag{1}$$

Pour la résolution $\Longrightarrow$ Pivot de Gauss.

Systèmes d'équations linéaires

On appelle système de m équations linéaires à n inconnues, un système de la forme :

$$
\begin{cases}
a_{11}x_1 + \cdots + a_{1j}x_j + \cdots + a_{1n}x_n = b_1 & L_1 \\
\vdots & \vdots \\
a_{i1}x_1 + \cdots + a_{ij}x_j + \cdots + a_{in}x_n = b_i & L_i \\
\vdots & \vdots \\
a_{m1}x_1 + \cdots + a_{mj}x_j + \cdots + a_{mn}x_n = b_m & L_m
\end{cases}
\tag{1}
$$

Pour la résolution $\Longrightarrow$ Pivot de Gauss.
Systèmes résolubles ?

Contenu

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{2}$$

se réécrit sous forme matricielle

$$\begin{pmatrix} 2 & 1 & -1 \\ -1 & 1 & -1 \\ 1 & 2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$$

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{2}$$

est équivalent à

$$\left(\begin{array}{ccc|c} 2 & 1 & -1 & 1 \\ -1 & 1 & -1 & -2 \\ 1 & -2 & -1 & 3 \end{array}\right)$$

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{2}$$

est équivalent à

$$\left(\begin{array}{ccc|c} 1 & -2 & -1 & 3 \\ -1 & 1 & -1 & -2 \\ 2 & 1 & -1 & 1 \end{array}\right)$$

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{2}$$

est équivalent à

$$\left(\begin{array}{ccc|c} 1 & -2 & -1 & 3 \\ 0 & -1 & -2 & 1 \\ 0 & 5 & 2 & -5 \end{array}\right)$$

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{2}$$

est équivalent à

$$\left(\begin{array}{ccc|c} 1 & -2 & -1 & 3 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & -8 & 0 \end{array}\right)$$

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{2}$$

est équivalent à

$$\left(\begin{array}{ccc|c} 1 & -2 & 0 & 3 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{array} \right)$$

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{2}$$

est équivalent à

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{array} \right)$$

$\longrightarrow$ le système est équivalent à $x = 1$, $y = -1$, $z = 0$.

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Remarque :
Ajouter une ligne à une autre en dessous est équivalent à multiplier par une matrice *triangulaire inférieure*, de diagonale 1 : $\lambda E^{i,j}$

$$\begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix} \times \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & b \\ \lambda a + c & \lambda b + d \end{pmatrix}$$

où $E^{i,j}_{k,k} = 1$, $E^{i,j}_{i,j} = 1$ et sinon 0.

Remarque :
Ajouter une ligne à une autre en dessous est équivalent à multiplier par une matrice *triangulaire inférieure*, de diagonale 1 : $\lambda E^{i,j}$ où $E^{i,j}_{k,k} = 1$, $E^{i,j}_{i,j} = 1$ et sinon 0.
Donc, via pivot de gauss, si on a pas besoin de permuter

$$A = L_1 \times L_2 \times \cdots \times L_k [\text{matrice triangulaire supérieure}] \tag{3}$$

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Remarque :
Ajouter une ligne à une autre en dessous est équivalent à multiplier par une matrice *triangulaire inférieure*, de diagonale 1 : $\lambda E^{i,j}$ où $E^{i,j}_{k,k} = 1$, $E^{i,j}_{i,j} = 1$ et sinon 0.
Donc, via pivot de gauss, si on a pas besoin de permuter

$$A = L_1 \times L_2 \times \cdots \times L_k [\text{matrice triangulaire supérieure}] \tag{3}$$

i.e. $A = LU$ avec L triangulaire inférieure ($L \longleftarrow$ “lower”) de diagonale 1, et U triangulaire supérieure ($U \longleftarrow$ “upper”).

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Remarque :
Ajouter une ligne à une autre en dessous est équivalent à multiplier par une matrice *triangulaire inférieure*, de diagonale 1 : $\lambda E^{i,j}$ où $E^{i,j}_{k,k} = 1$, $E^{i,j}_{i,j} = 1$ et sinon 0.
Donc, via pivot de gauss, si on a pas besoin de permuter

$$A = L_1 \times L_2 \times \cdots \times L_k [\text{matrice triangulaire supérieure}] \qquad (3)$$

i.e. $A = LU$ avec L triangulaire inférieure ($L \leftarrow$ "lower") de diagonale 1, et U triangulaire supérieure ($U \leftarrow$ "upper").
En pratique :
Utiliser un algo (optimisé) déjà implémenté. Complexité : $O(n^3)$.

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Autre remarque

Si on connaît un x_1, x_2, et x_3 tels que

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Autre remarque

Si on connaît un x_1, x_2, et x_3 tels que

$$Ax_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad Ax_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad Ax_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Autre remarque

Si on connaît un x_1, x_2, et x_3 tels que

$$Ax_1 = \begin{pmatrix}1\\0\\0\end{pmatrix}, \quad Ax_2 = \begin{pmatrix}0\\1\\0\end{pmatrix}, \quad Ax_3 = \begin{pmatrix}0\\0\\1\end{pmatrix}.$$

Alors, pour n'importe quel $b \in \mathbb{R}^3$,

$$Ax_b := A(b_1x_1 + b_2x_2 + b_3x_3) = b_1Ax_1 + b_2Ax_2 + b_3Ax_3 = b.$$

$\longrightarrow$ **Quand ça existe** on appelle $A^{-1} := (x_1|x_2|x_3)$ l'inverse de A.

Aparté. Pivot de Gauss et décomposition $(P)LU$.

Autre remarque
Si on connaît un x_1, x_2, et x_3 tels que

$$Ax_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad Ax_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad Ax_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Alors, pour n'importe quel $b \in \mathbb{R}^3$,

$$Ax_b := A(b_1x_1 + b_2x_2 + b_3x_3) = b_1Ax_1 + b_2Ax_2 + b_3Ax_3 = b.$$

$\longrightarrow$ **Quand ça existe** on appelle $A^{-1} := (x_1|x_2|x_3)$ l'inverse de A.
Remarque : $A = PLU \implies P, L$ inversible, et U est inversible ssi ses coeffs diagonaux sont non nuls.

Contenu

Équations linéaires ⟶ Algèbre linéaire

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{4}$$

se réécrit en

$$\begin{pmatrix} 2 & 1 & -1 \\ -1 & 1 & -1 \\ 1 & 2 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix}$$

qui se réécrit en général

$$AX = b$$

où

$$A = (a_{i,j})_{1 \leq i,j \leq n}, \quad X = (x_1, \ldots, x_n)^T, \text{ et } b = (b_i)^T_{1 \leq i \leq n}.$$

Équations linéaires ⟶ Algèbre linéaire

Le système :

$$\begin{cases} 2x + y - z = 1 & L_1 \\ -x + y - z = -2 & L_2 \\ x - 2y - z = 3 & L_3 \end{cases} \tag{4}$$

qui se réécrit en général

$$AX = b$$

où

$$A = (a_{i,j})_{1 \leq i,j \leq n}, \quad X = (x_1, \ldots, x_n)^T, \text{ et } b = (b_i)^T_{1 \leq i \leq n}.$$

$\longrightarrow$ Ce qui compte c'est donc comprendre comment les applications $x \mapsto Ax$ se comportent.

Petit raccourci

L'identification :

$$A \text{ de dimension } n \times m \quad \longleftrightarrow \quad x \in \mathbb{R}^m \longmapsto Ax \in \mathbb{R}^n$$

permet de faire passer la théorie des fonctions aux matrices.

Petit raccourci

L'identification :

$$A \text{ de dimension } n \times m \quad \longleftrightarrow \quad x \in \mathbb{R}^m \longmapsto Ax \in \mathbb{R}^n$$

permet de faire passer la théorie des fonctions aux matrices.

Exemple : injectivité, surjectivité, continuité, différentiabilité, inversibilité …

Petit raccourci

L'identification :

$$A \text{ de dimension } n \times m \quad \longleftrightarrow \quad x \in \mathbb{R}^m \longmapsto Ax \in \mathbb{R}^n$$

permet de faire passer la théorie des fonctions aux matrices.

Exemple : injectivité, surjectivité, continuité, différentiabilité, inversibilité …

Remarque : $x \mapsto A^{-1}x$ défini plus haut est l'inverse de $x \mapsto Ax$.

Autre remarque anodine mais pas trop

Une matrice est caractérisée par son image des vecteurs $e_i = (0, \dots, 1, \dots, 0)^T$.

$$Ae_i = \begin{pmatrix} a_{1i} \\ \vdots \\ a_{ni} \end{pmatrix} = \mathrm{Col}_i(A)$$

Autre remarque anodine mais pas trop

Une matrice est caractérisée par son image des vecteurs $e_i = (0, \ldots, 1, \ldots, 0)^T$.

$$Ae_i = \begin{pmatrix} a_{1i} \\ \vdots \\ a_{ni} \end{pmatrix} = \mathrm{Col}_i(A)$$

En particulier, pour toute application linéaire (de dimension finie) *i.e.* application $f : x \in \mathbb{R}^n \longmapsto f(x) \in \mathbb{R}^m$ qui vérifie

$$\forall x, y \in \mathbb{R}^n, \lambda \in \mathbb{R}, \quad f(x + \lambda y) = f(x) + \lambda f(y)$$

il existe une matrice A_f de dimension $n \times m$ telle que $f = x \mapsto A_f x$.

Injectivité ?

En remarquant

$$Ax = Ay \Leftrightarrow A(x - y) = 0$$

Injectivité ?

En remarquant

$$Ax = Ay \Leftrightarrow A(x - y) = 0$$

$\longrightarrow$ injectivité caractérisée par les x tels que

$$Ax = \mathbf{0} = (0, \ldots, 0)^T$$

Injectivité ?

En remarquant

$$Ax = Ay \Leftrightarrow A(x - y) = 0$$

$\longrightarrow$ injectivité caractérisée par les x tels que

$$Ax = \mathbf{0} = (0, \ldots, 0)^T$$

D'où la définition du *noyau* (*kern* en Allemand) de A

$$\ker A = \{x \in \mathbb{R}^n : Ax = \mathbf{0}\} = A^{-1}\{\mathbf{0}\}$$

Injectivité ?

En remarquant

$$Ax = Ay \Leftrightarrow A(x - y) = 0$$

$\longrightarrow$ injectivité caractérisée par les x tels que

$$Ax = \mathbf{0} = (0, \ldots, 0)^T$$

D'où la définition du *noyau* (*kern* en Allemand) de A

$$\ker A = \{x \in \mathbb{R}^n : Ax = \mathbf{0}\} = A^{-1}\{\mathbf{0}\}$$

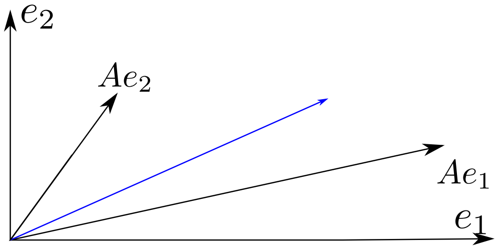
Calcul ?

$\Longrightarrow$ Pareil que pour résoudre un système $Ax = b$ avec $b = \mathbf{0}$: Gauss.

Contenu

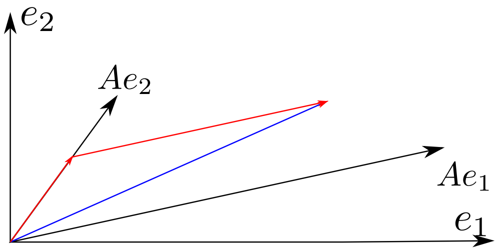
Raccourci théorique. Dimension 2.

À quel point une matrice est proche d'être inversible ? Un indicateur :



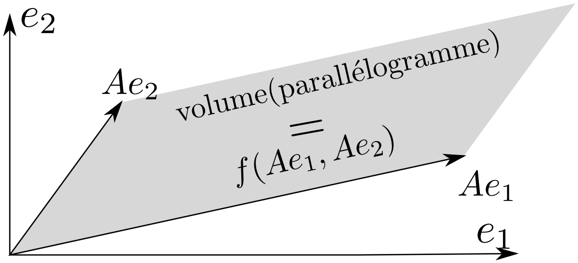
Raccourci théorique. Dimension 2.

À quel point une matrice est proche d'être inversible ? Un indicateur :



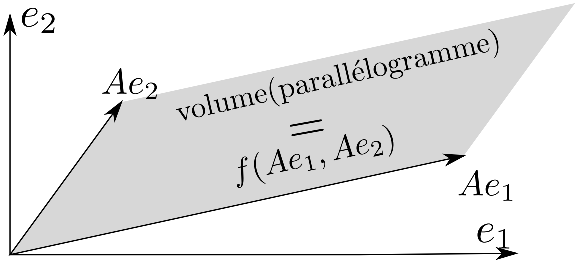
Raccourci théorique. Dimension 2.

À quel point une matrice est proche d'être inversible ? Un indicateur :



Raccourci théorique. Dimension 2.

À quel point une matrice est proche d'être inversible ? Un indicateur :



On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.

Raccourci théorique. Dimension 2.

On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.
Ça impose :

$$f(x+y, x+y) = f(x, x+y) + f(y, x+y)$$

Raccourci théorique. Dimension 2.

On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.

$$\begin{aligned} f(x+y, x+y) &= f(x, x+y) + f(y, x+y) \\ &= f(x, x) + f(y, y) + f(y, x) + f(x, y) \end{aligned}$$

Donc $f(x, y) = -f(y, x)$.

Raccourci théorique. Dimension 2.

On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.
Donc $f(x, y) = -f(y, x)$.
Compatibilité avec Gauss (version colonne) :

- Ajouter une colonne à une autre :

$$f(x, \lambda x + y) = \lambda f(x, x) + f(x, y) = f(x, y)$$

- Permuter deux colonnes : on perd un -1.

Raccourci théorique. Dimension 2.

On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.
Donc $f(x, y) = -f(y, x)$.
$\longrightarrow$ Faire du Gauss ne change pas f à un signe près.

Raccourci théorique. Dimension 2.

On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.
Donc $f(x, y) = -f(y, x)$.
$\longrightarrow$ Faire du Gauss ne change pas f à un signe près.
$\longrightarrow$ On se ramène via Gauss à

$$f(A) = \pm f\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix} = \pm a_1 \cdot a_2 f(I_2) = \pm a_1 a_2$$

où a_1 et a_2 sont les coeffs diagonaux de U dans la décompo PLU de A; si on impose $f(I_2) = \text{aire}([0, 1]^2) = 1$.

Raccourci théorique. Dimension 2.

On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.
Donc $f(x, y) = -f(y, x)$.
$\longrightarrow$ Faire du Gauss ne change pas f à un signe près.
$\longrightarrow$ On se ramène via Gauss à

$$f(A) = \pm f\begin{pmatrix} a_1 & 0 \\ 0 & a_2 \end{pmatrix} = \pm a_1 \cdot a_2 f(I_2) = \pm a_1 a_2$$

où a_1 et a_2 sont les coeffs diagonaux de U dans la décompo PLU de A; si on impose $f(I_2) = \text{aire}([0, 1]^2) = 1$.
$\longrightarrow$ f est caractérisée par : $f(x, x) = 0$ et $f(\text{carré}) = 1$.

Raccourci théorique. Dimension 2.

On veut une fonction $f(x, y)$ qui soit linéaire en x et en y et telle que $f(x, x) = 0$.
Donc $f(x, y) = -f(y, x)$.
$\longrightarrow$ Faire du Gauss ne change pas f à un signe près.
$\longrightarrow$ f est caractérisée par : $f(x, x) = 0$ et $f(\text{carré}) = 1$.
Remarque : $f(A) \neq 0 \Leftrightarrow A$ inversible.

Raccourci théorique. Dimension quelconque.

En général (dimension ≥ 2) si :

$$x_i = x_j \implies f(x_1, x_2, \ldots, x_n) = 0, \qquad \text{et } f(I_n) = \mathrm{vol}([0,1]^n) = 1$$

Raccourci théorique. Dimension quelconque.

En général (dimension ≥ 2) si :

$$x_i = x_j \implies f(x_1, x_2, \ldots, x_n) = 0, \qquad \text{et } f(I_n) = \mathrm{vol}([0,1]^n) = 1$$

alors :

$$A = PLU \implies f(A) = f(PLU) = \varepsilon(P) f(LU) = \varepsilon(P) f(U) = \varepsilon(P) a_1 \cdots a_n$$

où $\varepsilon(P)$ est la signature de la permutation P, et $(a_i)_i$ les coeffs diagonaux de U.

Raccourci théorique. Dimension quelconque.

En général (dimension ≥ 2) si :

$$x_i = x_j \implies f(x_1, x_2, \ldots, x_n) = 0, \qquad \text{et } f(I_n) = \mathrm{vol}([0,1]^n) = 1$$

alors :

$$A = PLU \implies f(A) = f(PLU) = \varepsilon(P) f(LU) = \varepsilon(P) f(U) = \varepsilon(P) a_1 \cdots a_n$$

où $\varepsilon(P)$ est la signature de la permutation P, et $(a_i)_i$ les coeffs diagonaux de U.

$$\implies f(A) = \det(A) = \frac{1}{n!} \sum_P \varepsilon(P) \det(PA) = \frac{1}{n!} \sum_{\sigma \in \mathfrak{S}_n} \varepsilon(\sigma) \prod_i a_{i,\sigma(i)}.$$

Raccourci théorique. Dimension quelconque.

En général (dimension ≥ 2) si :

$$x_i = x_j \implies f(x_1, x_2, \dots, x_n) = 0, \qquad \text{et } f(I_n) = \mathrm{vol}([0,1]^n) = 1$$

alors :

$$A = PLU \implies f(A) = f(PLU) = \varepsilon(P) f(LU) = \varepsilon(P) f(U) = \varepsilon(P) a_1 \cdots a_n$$

où $\varepsilon(P)$ est la signature de la permutation P, et $(a_i)_i$ les coeffs diagonaux de U.

$$\implies f(A) = \det(A) = \frac{1}{n!} \sum_P \varepsilon(P) \det(PA) = \frac{1}{n!} \sum_{\sigma \in \mathfrak{S}_n} \varepsilon(\sigma) \prod_i a_{i,\sigma(i)}.$$

En pratique : Utiliser un algo déjà implémenté. Complexité : $O(n^3)$.

Contenu

Famille libres, génératrices, dimension

$(x_1, \ldots, x_k)$ famille libre $\Leftrightarrow$ $(x_1 | \cdots | x_k)$ matrice injective.

Famille libres, génératrices, dimension

$(x_1, \dots, x_k)$ famille libre $\Leftrightarrow (x_1 | \cdots | x_k)$ matrice injective.
$(x_1, \dots, x_k)$ famille génératrice $\Leftrightarrow (x_1 | \cdots | x_k)$ matrice surjective.

Famille libres, génératrices, dimension

$(x_1, \ldots, x_k)$ famille libre $\Leftrightarrow$ $(x_1| \cdots |x_k)$ matrice injective.
$(x_1, \ldots, x_k)$ famille génératrice $\Leftrightarrow$ $(x_1| \cdots |x_k)$ matrice surjective.
$(x_1, \ldots, x_n)$ base de $\mathbb{R}^n$ $\Leftrightarrow$ $(x_1, \ldots, x_n)$ libre et génératrice de $\mathbb{R}^n$.

Famille libres, génératrices, dimension

$(x_1, \dots, x_k)$ famille libre $\Leftrightarrow$ $(x_1 | \cdots | x_k)$ matrice injective.
$(x_1, \dots, x_k)$ famille génératrice $\Leftrightarrow$ $(x_1 | \cdots | x_k)$ matrice surjective.
$(x_1, \dots, x_n)$ base de $\mathbb{R}^n$ $\Leftrightarrow$ $(x_1, \dots, x_n)$ libre et génératrice de $\mathbb{R}^n$.
Dimension d'une famille : dimension de ce qui reste après Gauss.

Changement de base

$$(x_1, \ldots, x_n) \text{ est une base} \Leftrightarrow P = (x_1|\cdots|x_n) \text{ inversible} \Leftrightarrow \det(P) \neq 0.$$

Changement de base

$(x_1, \ldots, x_n)$ est une base $\Leftrightarrow P = (x_1|\cdots|x_n)$ inversible $\Leftrightarrow \det(P) \neq 0$.
Donc si A est de dimension $n \times n$:

$$Ae_i = (AP^{-1})(Pe_i) = (AP^{-1})x_i$$

Changement de base

$(x_1, \ldots, x_n)$ est une base $\Leftrightarrow P = (x_1 | \cdots | x_n)$ inversible $\Leftrightarrow \det(P) \neq 0$.
Donc si A est de dimension $n \times n$:

$$Ae_i = (AP^{-1})(Pe_i) = (AP^{-1})x_i$$

Donc $x \mapsto AP^{-1}x$ est caractérisée par son image sur les x_i, donc $x \mapsto Ax$ idem.

Changement de base

$(x_1, \ldots, x_n)$ est une base $\Leftrightarrow P = (x_1 | \cdots | x_n)$ inversible $\Leftrightarrow \det(P) \neq 0$.
Donc si A est de dimension $n \times n$:

$$Ae_i = (AP^{-1})(Pe_i) = (AP^{-1})x_i$$

Donc $x \mapsto AP^{-1}x$ est caractérisée par son image sur les x_i, donc $x \mapsto Ax$ idem.
Plus précisément, si $Ae_i = \mathrm{Col}_i(A) = \sum_{j=1}^{n} \lambda_j e_j$, on a

$$PAP^{-1}x_i = PAe_i = P\left(\sum_j \lambda_j e_j\right) = \sum_j \lambda_j x_j.$$

Donc, si $e_i \leftrightarrow x_i$ alors $\mathrm{Col}_i(A) \leftrightarrow \mathrm{Col}_i(A)$ « dans la base x_i ».

Valeur propre, vecteur propre.

A matrice $n \times n$.

$$\lambda \in \mathbb{C} \text{ valeur propre} \Leftrightarrow \det(A - \lambda I_n) = 0$$

$$\Updownarrow$$

$$A - \lambda I_n \text{ non inversible.}$$

Valeur propre, vecteur propre.

A matrice $n \times n$.

$$\begin{array}{rcl} \lambda \in \mathbb{C} \text{ valeur propre} & \Leftrightarrow & \det(A - \lambda I_n) = 0 \\ & & \Updownarrow \\ & & A - \lambda I_n \text{ non inversible.} \end{array}$$

$$\begin{array}{rcl} x \in \mathbb{C}^n \text{ vecteur propre associé à } \lambda & \Leftrightarrow & Ax = \lambda x \\ & & \Updownarrow \\ x \in E_\lambda & \Leftrightarrow & x \in \ker(A - \lambda I_n) \end{array}$$

Polynôme caractéristique

Polynôme caractéristique de A :

$$\chi_A(X) = \det(A - XI_n)$$

Polynôme caractéristique

Polynôme caractéristique de A :

$$\chi_A(X) = \det(A - XI_n)$$

$$\begin{array}{ccc} \lambda \in \mathbb{C} \text{ valeur propre} & \Leftrightarrow & \det(A - \lambda I_n) = 0 \\ \Updownarrow & & \Updownarrow \\ \chi_A(\lambda) = 0 & \Leftrightarrow & A - \lambda I_n \text{ non inversible.} \end{array}$$

Diagonalisation

Une matrice est diagonalisable si elle est diagonale dans une base de vecteurs propres.

$$PAP^{-1} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix}$$

Diagonalisation

Une matrice est diagonalisable si elle est diagonale dans une base de vecteurs propres.

$$PAP^{-1} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix}$$

Diagonalisabilité ?

Arguments techniques pour la diagonalisation

$\longrightarrow$ Théorème de d'Alembert-Gauss :

$$\chi_A(X) = \sum_{i=0}^{n} a_i X^i = \prod_{\lambda} (X - \lambda)^{m_\lambda}$$

Arguments techniques pour la diagonalisation

$\longrightarrow$ Théorème de d'Alembert-Gauss :

$$\chi_A(X) = \sum_{i=0}^{n} a_i X^i = \prod_{\lambda} (X - \lambda)^{m_\lambda}$$

$\longrightarrow$ Théorème de Cayley-Hamilton :

$$\chi_A(A) = \sum_{i=0}^{n} a_i A^i = \prod_{\lambda} (A - \lambda I_n)^{m_\lambda} = 0$$

Arguments techniques pour la diagonalisation

$\longrightarrow$ Théorème de d'Alembert-Gauss :

$$\chi_A(X) = \sum_{i=0}^{n} a_i X^i = \prod_{\lambda} (X - \lambda)^{m_\lambda}$$

$\longrightarrow$ Théorème de Cayley-Hamilton :

$$\chi_A(A) = \sum_{i=0}^{n} a_i A^i = \prod_{\lambda} (A - \lambda I_n)^{m_\lambda} = 0$$

$\longrightarrow$ Lemme des noyaux :

$$\mathbb{R}^n = \ker(\chi_A(A)) = \bigoplus_{\lambda} \ker\left[(A - \lambda I)^{m_\lambda}\right] = \bigoplus_{\lambda} E_\lambda$$

Conditions suffisantes

1. Théorème spectral

Conditions suffisantes pour la diagonalisation

Conditions suffisantes

- Toutes les valeurs propres sont distinctes.

1. Théorème spectral

Conditions suffisantes pour la diagonalisation

Conditions suffisantes

- Toutes les valeurs propres sont distinctes.
- Matrice symétrique (avec des valeurs propres réelles [1]).

Conditions équivalentes

1. Théorème spectral

Conditions suffisantes pour la diagonalisation

Conditions suffisantes

- Toutes les valeurs propres sont distinctes.
- Matrice symétrique (avec des valeurs propres réelles [1]).

Conditions équivalentes

- Polynôme minimal scindé. (pas défini ici)

1. Théorème spectral

Conditions suffisantes pour la diagonalisation

Conditions suffisantes

- Toutes les valeurs propres sont distinctes.
- Matrice symétrique (avec des valeurs propres réelles [1]).

Conditions équivalentes

- Polynôme minimal scindé. (pas défini ici)
- $\ker(A - \lambda I)^{m_\lambda}$ est composé seulement de vecteurs propres.

1. Théorème spectral

Conditions suffisantes pour la diagonalisation

Conditions suffisantes

- Toutes les valeurs propres sont distinctes.
- Matrice symétrique (avec des valeurs propres réelles [1]).

Conditions équivalentes

- Polynôme minimal scindé. (pas défini ici)
- $\ker(A - \lambda I)^{m_\lambda}$ est composé seulement de vecteurs propres.

En pratique : Une matrice est *presque sûrement* diagonalisable, on utilise un algo déjà implémenté.

1. Théorème spectral

Contenu

Norme d'un vecteur

Pour $x \in \mathbb{R}^n$.
90% du temps : Norme Euclidienne

$$\|x\|_2 = \|(x_1, \ldots, x_n)\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{\frac{1}{2}}.$$

Norme d’un vecteur

Pour $x \in \mathbb{R}^n$.
90% du temps : Norme Euclidienne

$$\|x\|_2 = \|(x_1, \dots, x_n)\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}.$$

99% du temps : Norme L^p. Pour $p \in [1, \infty)$

$$\|x\|_p = \|(x_1, \dots, x_n)\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}.$$

ou $p = \infty$: $\|x\|_\infty = \max\{x_1, \dots, x_n\}$.

En image

Animation

Définition / Propriétés d'une norme

- (Séparation) $\|x\| = 0 \Leftrightarrow x = 0$,

Définition / Propriétés d'une norme

- (Séparation) $\|x\| = 0 \Leftrightarrow x = 0$,
- (Inégalité triangulaire) $|\|x\| - \|y\|| \leq \|x + y\| \leq \|x\| + \|y\|$,

Définition / Propriétés d'une norme

- (Séparation) $\|x\| = 0 \Leftrightarrow x = 0$,
- (Inégalité triangulaire) $|\|x\| - \|y\|| \leq \|x + y\| \leq \|x\| + \|y\|$,
- (Homogénéité) $\|ax\| = |a|\|x\|$,

Définition / Propriétés d'une norme

- (Séparation) $\|x\| = 0 \Leftrightarrow x = 0$,
- (Inégalité triangulaire) $|\|x\| - \|y\|| \leq \|x + y\| \leq \|x\| + \|y\|$,
- (Homogénéité) $\|ax\| = |a|\|x\|$,

- (Positivité) $\|x\| \geq 0$,

Définition / Propriétés d'une norme

- (Séparation) $\|x\| = 0 \Leftrightarrow x = 0$,
- (Inégalité triangulaire) $|\|x\| - \|y\|| \leq \|x + y\| \leq \|x\| + \|y\|$,
- (Homogénéité) $\|ax\| = |a|\|x\|$,

- (Positivité) $\|x\| \geq 0$,
- $\| - x\| = \|x\|$.

Définition / Propriétés d'une norme

- (Séparation) $\|x\| = 0 \Leftrightarrow x = 0$,
- (Inégalité triangulaire) $|\|x\| - \|y\|| \leq \|x + y\| \leq \|x\| + \|y\|$,
- (Homogénéité) $\|ax\| = |a|\|x\|$,

- (Positivité) $\|x\| \geq 0$,
- $\|-x\| = \|x\|$.

- Toutes les normes sont équivalentes.

Norme de matrice / d'*opérateur* (ou *subordonnées*)

Normes issue de : matrice = tableau de nombres = gros vecteur

- Norme de Frobenius :

$$||A||_F = \sqrt{\sum_i \sum_j A_{i,j}^2} = \sqrt{\mathrm{trace}(A^* A)}$$

Norme de matrice / d'*opérateur* (ou *subordonnées*)

Normes issue de : matrice = tableau de nombres = gros vecteur

- Norme de Frobenius :

$$\|A\|_F = \sqrt{\sum_i \sum_j A_{i,j}^2} = \sqrt{\mathrm{trace}(A^* A)}$$

- Normes L^p :

$$\|A\|_\infty = \max_{i,j} \left|A_{i,j}\right|, \quad \|A\|_{L^p} = \left(\sum_i \sum_j \left|A_{i,j}\right|^p\right)^{\frac{1}{p}}$$

Norme de matrice / d'*opérateur* (ou *subordonnées*)

Normes issue de : matrice = tableau de nombres = gros vecteur

- Norme de Frobenius :

$$\|A\|_F = \sqrt{\sum_i \sum_j A_{i,j}^2} = \sqrt{\mathrm{trace}(A^*A)}$$

- Normes L^p :

$$\|A\|_\infty = \max_{i,j} \left|A_{i,j}\right|, \quad \|A\|_{L^p} = \left(\sum_i \sum_j \left|A_{i,j}\right|^p\right)^{\frac{1}{p}}$$

Normes issues de l'identification : $x \in \mathbb{R}^m \longmapsto Ax \in \mathbb{R}^n$.
Idée : «taille» de Ax VS «taille» de x ?

Norme de matrice / d'*opérateur* (ou *subordonnées*)

Normes issue de : matrice = tableau de nombres = gros vecteur

- Norme de Frobenius :

$$\|A\|_F = \sqrt{\sum_i \sum_j A_{i,j}^2} = \sqrt{\mathrm{trace}(A^* A)}$$

- Normes L^p :

$$\|A\|_\infty = \max_{i,j} \left|A_{i,j}\right|, \quad \|A\|_{L^p} = \left(\sum_i \sum_j \left|A_{i,j}\right|^p\right)^{\frac{1}{p}}$$

Normes issues de l'identification : $x \in \mathbb{R}^m \longmapsto Ax \in \mathbb{R}^n$.
Idée : «taille» de Ax VS «taille» de x ? *i.e.*

$$\|Ax\|_{\mathbb{R}^n} \quad \text{VS} \quad \|x\|_{\mathbb{R}^m}?$$

Norme de matrice / d'*opérateur* (ou *subordonnées*)

Normes issue de : matrice = tableau de nombres = gros vecteur

- Norme de Frobenius :

$$\|A\|_F = \sqrt{\sum_i \sum_j A_{i,j}^2} = \sqrt{\mathrm{trace}(A^* A)}$$

- Normes L^p :

$$\|A\|_\infty = \max_{i,j} \left|A_{i,j}\right|, \quad \|A\|_{L^p} = \left(\sum_i \sum_j \left|A_{i,j}\right|^p\right)^{\frac{1}{p}}$$

Normes issues de l'identification : $x \in \mathbb{R}^m \longmapsto Ax \in \mathbb{R}^n$.
Idée : «taille» de Ax VS «taille» de x ? *i.e.*

$$\|Ax\|_{\mathbb{R}^n} \quad \text{VS} \quad \|x\|_{\mathbb{R}^m} ?$$

$$\longrightarrow \|A\| = \|x \mapsto Ax\| = \sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|\leq 1} \frac{\|Ax\|}{\|x\|}$$

Propriétés

- Subordonnées ou Frobenius $\implies$ norme d'algèbre *i.e.*

$$\|AB\| \leq \|A\| \cdot \|B\|,$$

Propriétés

- Subordonnées ou Frobenius $\implies$ norme d'algèbre *i.e.*

$$\|AB\| \leq \|A\| \cdot \|B\|,$$

- Cas spéciaux des normes subordonnées, avec A diagonalisable :

$$\|A\|_1 = \max_j \sum_i |a_{i,j}|, \quad \|A\|_\infty = \max_i \sum_j |a_{i,j}|, \quad \|A\|_2 = |\lambda_{\max}(A)|$$

Propriétés

- Subordonnées ou Frobenius $\implies$ norme d'algèbre *i.e.*

$$\|AB\| \leq \|A\| \cdot \|B\|,$$

- Cas spéciaux des normes subordonnées, avec A diagonalisable :

$$\|A\|_1 = \max_j \sum_i |a_{i,j}|, \quad \|A\|_\infty = \max_i \sum_j |a_{i,j}|, \quad \|A\|_2 = |\lambda_{\max}(A)|$$

- Pour Frobenius,

$$\|A\|_F = \sqrt{\text{trace}(A'A)} = \sqrt{\lambda_1^2 + \cdots + \lambda_n^2}$$

Contenu

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

- Pas d'hypothèses pour l'existence,

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

- Pas d'hypothèses pour l'existence,
- Marche sur des matrices pas carrées,

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

- Pas d'hypothèses pour l'existence,
- Marche sur des matrices pas carrées,
- Même complexité (algo) en $O(n^3)$,

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

- Pas d'hypothèses pour l'existence,
- Marche sur des matrices pas carrées,
- Même complexité (algo) en $O(n^3)$,
- Pratique en pratique.

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

- Pas d'hypothèses pour l'existence,
- Marche sur des matrices pas carrées,
- Même complexité (algo) en $O(n^3)$,
- Pratique en pratique.

$\longrightarrow$ Moins bien :

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

- Pas d'hypothèses pour l'existence,
- Marche sur des matrices pas carrées,
- Même complexité (algo) en $O(n^3)$,
- Pratique en pratique.

$\longrightarrow$ Moins bien :

- 2 changements de bases au lieu d'un,

Comparaison avec la diagonalisation classique

$\longrightarrow$ Bien :

- Pas d'hypothèses pour l'existence,
- Marche sur des matrices pas carrées,
- Même complexité (algo) en $O(n^3)$,
- Pratique en pratique.

$\longrightarrow$ Moins bien :

- 2 changements de bases au lieu d'un,
- Valeurs *singulières* au lieu de valeurs propres.

SVD : Comment ça marche.

Décomposition polaire

Toute matrice M est le produit d'une matrice symétrique positive S et une matrice orthogonale O (*i.e.* $O^T O = I_n$). $M = SO$.

$$z \leftrightarrow re^{i\theta} \in \mathbb{C}.$$

Décomposition polaire
Toute matrice M est le produit d'une matrice symétrique positive S et une matrice orthogonale O (*i.e.* $O^T O = I_n$). $M = SO$.

$$z \leftrightarrow re^{i\theta} \in \mathbb{C}.$$

Idée (cas M inversible)
$M \longrightarrow M'M$ symétrique positive donc il existe une matrice orthogonale U telle que

$$M'M = U \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} U^{-1}$$

avec des $\lambda_i > 0$.

SVD : Comment ça marche.

Décomposition polaire
Toute matrice M est le produit d'une matrice symétrique positive S et une matrice orthogonale O (*i.e.* $O^T O = I_n$). $M = SO$.

$$z \leftrightarrow re^{i\theta} \in \mathbb{C}.$$

Idée (cas M inversible)
$M \longrightarrow M'M$ symétrique positive donc il existe une matrice orthogonale U telle que

$$M'M = U \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} U^{-1}$$

avec des $\lambda_i > 0$.

$$S := U \begin{pmatrix} \sqrt{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_n} \end{pmatrix} U^{-1}$$

SVD : Comment ça marche.

Idée (cas M inversible)
$M \longrightarrow M'M$ symétrique positive donc il existe une matrice orthogonale U telle que

$$M'M = U \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} U^{-1}$$

avec des $\lambda_i > 0$.

$$S := U \begin{pmatrix} \sqrt{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_n} \end{pmatrix} U^{-1}$$

$\longrightarrow$ On vérifie que $O := S^{-1}M$ est orthogonale et $M = SO$.

SVD : Comment ça marche.

Idée (cas M inversible)
$M \longrightarrow M'M$ symétrique positive donc il existe une matrice orthogonale U telle que

$$M'M = U \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} U^{-1}$$

avec des $\lambda_i > 0$.

$$S := U \begin{pmatrix} \sqrt{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_n} \end{pmatrix} U^{-1}$$

$\longrightarrow$ On vérifie que $O := S^{-1}M$ est orthogonale et $M = SO$.
$\longrightarrow$ Si M n'est pas inversible : densité de $\mathrm{GL}_n(\mathbb{C})$ dans $\mathcal{M}_n(\mathbb{C})$.

SVD : Comment ça marche.

$$\begin{aligned}
M \longrightarrow & M = SO && \text{(Décomposition polaire)} \\
\longrightarrow & S = UDU^{-1} && \text{(Théorème spectral)} \\
\longrightarrow & M = UDU^{-1}O = (U) \times (D) \times \left(U^{-1}O\right).
\end{aligned}$$

SVD : Comment ça marche.

$$
\begin{aligned}
M \longrightarrow & M = SO && \text{(Décomposition polaire)} \\
\longrightarrow & S = UDU^{-1} && \text{(Théorème spectral)} \\
\longrightarrow & M = UDU^{-1}O = (U) \times (D) \times \left(U^{-1}O\right).
\end{aligned}
$$

Remarque : D est positive.

SVD : Comment ça marche.

$$
\begin{aligned}
M \longrightarrow & M = SO && \text{(Décomposition polaire)}\\
\longrightarrow & S = UDU^{-1} && \text{(Théorème spectral)}\\
\longrightarrow & M = UDU^{-1}O = (U) \times (D) \times \left(U^{-1}O\right).
\end{aligned}
$$

Remarque : D est positive.
En pratique : On utilise un algo optimisé déjà implémenté.
Complexité $O(n^3)$.

Contenu

Application SVD.