Álgebra Lineal Computacional

Clase 1 - 16 / agosto / 2022

Segundo Cuatrimestre 2022

Facultad de Ciencias Exacas y Naturales, UBA

Vectores

Para $n \in \mathbb{N}$, definimos un vector de n coordenadas como una sucesión de n números reales o complejos $\boldsymbol{v} = (v_1, \ldots, v_n)$.

- $\mathbb{R}^n$ es el conjunto de todos los vectores reales de n coordenadas
- $\mathbb{C}^n$ al conjunto de todos los vectores complejos de n coordenadas.

Nos referiremos a este conjunto como $\mathbb{K}$ cuando los resultados valgan tanto para $\mathbb{R}$ como para $\mathbb{C}$.

Ejemplo

- $\mathbb{R}^2$ es el plano coordenado.
- $\mathbb{R}^3$ es el espacio de 3 dimensiones.

Vectores

Ejemplo

- $v = (1 - 2i, 2 + 3i) \in \mathbb{C}^2$.
- $u = (2, -1, \pi, 3/2) \in \mathbb{R}^4$.

En `Python` definimos vectores con el comando `array` (arreglo) del paquete `numpy`.

```
import numpy as np
v1 = np.array([10, 5, -7, 1])
print(v1)

v2 = np.array([5, 0, 3/2, 2])
print(v2)
```

Matrices

Denotamos $\mathbb{K}^{m\times n}$ al espacio de matrices de números reales o complejos de m filas y n columnas, que podemos considerar como un vector de $m \times n$ coordenadas agrupadas en m filas de n coordenadas.

Ejemplo

1. $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \in \mathbb{R}^{2\times 3}$
2. $\begin{pmatrix} 1 \\ 7 \\ 0.6 \\ -1 \end{pmatrix} \in \mathbb{R}^{4\times 1}$
3. $\begin{pmatrix} 1 & 5+i \\ 7-2i & 0 \\ i & 3 \end{pmatrix} \in \mathbb{C}^{3\times 2}$

Es común considerar a los vectores en $\mathbb{K}^n$ como matrices columna. Es decir, al vector $v = (1, 2, 3)$ lo pensamos como matriz $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.

Matrices

En `Python` definimos matrices también con `array`, como arreglo de arreglos, es decir como un arreglo de m vectores, donde cada vector es una fila.

```
A = np.array([[1,2],[3,4]]) # Matriz de 2 x 2
print("A = \n", A)
B = np.array([[1,2,3,4],[7,1,2,-1]]) # Matriz de 4 x 2
print("B = \n", B)
C = np.array([[1], [7], [1/3]]) # Matriz columna de 1 x 3
print("C = \n", C)
```

Sistemas lineales de ecuaciones

Resolución de sistemas de ecuaciones por "triangulación"

La *eliminación gaussiana* es un método muy eficiente para resolver sistemas de ecuaciones lineales en forma directa (es decir, sin utilizar métodos iterativos que aproximan la solución).

A modo de ejemplo, resolvemos el siguiente sistema de ecuaciones.

$$\begin{cases} x + 5y + 5z = 2 \\ 2x + 2y - 3z = -1 \\ -x - 9y + 2z = 9. \end{cases}$$

Sistemas lineales de ecuaciones

Matriz ampliada

A partir del sistema, construimos la **matriz ampliada** de coeficientes y términos independientes:

$$\left(\begin{array}{rrr|r} 1 & 5 & 5 & 2 \\ 2 & 2 & -3 & -1 \\ -1 & -9 & 2 & 9 \end{array}\right).$$

Triangulamos la matriz realizando operaciones de filas. Las operaciones permitidas (que no afectan las soluciones del sistema) son:

- sumarle o restarle a una fila un múltiplo de otra fila,
- intercambiar dos filas entre si.
- multiplicar una fila por un escalar distinto de 0.

Matriz escalonada

El algoritmo de eliminación gaussiana consiste en triangular o *escalonar* la matriz obteniendo 0's abajo de los elementos de la diagonal, o más generalmente, produciendo que en cada fila la cantidad de 0's iniciales sea mayor que en la fila anterior.

Realizamos las siguientes operaciones:

$$\left(\begin{array}{ccc|c} 1 & 5 & 5 & 2 \\ 2 & 2 & -3 & -1 \\ -1 & -9 & 2 & 9 \end{array}\right) \xrightarrow{f_2-2f_1\to f_2} \left(\begin{array}{ccc|c} 1 & 5 & 5 & 2 \\ 0 & -8 & -13 & -5 \\ -1 & -9 & 2 & 9 \end{array}\right) \to$$

$$\xrightarrow{f_3+f_1\to f_3} \left(\begin{array}{ccc|c} 1 & 5 & 5 & 2 \\ 0 & -8 & -13 & -5 \\ 0 & -4 & 7 & 11 \end{array}\right) \xrightarrow{f_3-\frac{1}{2}f_2\to f_3} \left(\begin{array}{ccc|c} 1 & 5 & 5 & 2 \\ 0 & -8 & -13 & -5 \\ 0 & 0 & \frac{27}{2} & \frac{27}{2} \end{array}\right)$$

Sustitución hacia atrás

A partir de la matriz escalonada obtenemos el sistema de ecuaciones

$$\begin{cases} x+5y+\ \ 5z = -1 \\ \ \ -8y-13z = -5 \\ \qquad \frac{27}{2}z = \frac{27}{2}, \end{cases}$$

que es equivalente al sistema original (tiene las mismas soluciones).

Ahora podemos obtener los valores de x, y, z por "sustitución hacia atrás":

- $\frac{27}{2}z = \frac{27}{2} \to z = 1.$
- $-8y - 13z = -5 \to y = \frac{-5+13z}{-8} = \frac{5-13}{8} = -1,$
- $x = -1 - 5y - 5z = -1.$

Matriz escalonada

Una matriz $A \in \mathbb{K}^{m \times n}$ se llama *escalonada* si para todo $2 \leq i \leq m$, la fila i tiene más ceros iniciales que la fila $i-1$.

Ejemplo

$$A = \begin{pmatrix} 1 & 2 & 0 & 7 & -1 \\ 0 & 1 & 2 & 4 & -1 \\ 0 & 0 & 0 & 0 & 5 \end{pmatrix}$$

es una matriz escalonada.

$$A = \begin{pmatrix} 1 & 2 & 0 & 7 & -1 \\ 0 & 0 & 0 & 4 & -1 \\ 0 & 0 & 0 & -3 & 5 \end{pmatrix}$$

no es una matriz escalonada.

Si a partir de la matriz ampliada, obtenemos una matriz escalonada equivalente, podemos resolver el sistema por sustitución hacia atras, despejando en cada fila la primera variable con coeficiente no nulo en función de las siguientes variables.

Eliminación gaussiana

Ejercicio

Resolver escalonando el sistema

$$\begin{cases} x_1 + 2x_2 = -1 \\ 2x_1 + 4x_2 + 3x_3 = 4 \\ 3x_3 = 6. \end{cases}$$

Construimos la matriz ampliada

$$\tilde{\boldsymbol{A}} = \left(\begin{array}{ccc|c} 1 & 2 & 0 & -1 \\ 2 & 4 & 3 & 4 \\ 0 & 0 & 3 & 6 \end{array}\right).$$

Matriz escalonada en Python

- En los paquetes comunes de Python no existe un comando para eliminación gaussiana.
- Utilizaremos el siguiente programa, extraido de la página https://math.stackexchange.com/questions/3073083/how-to-reduce-matrix-into-row-echelon-form-in-python/3073117.
- El nombre de la función es row_echelon, que es el nombre en inglés para una matriz escalonada por filas.

Matriz escalonada en Python

Probamos el programa en el siguiente ejemplo.

```
from row_echelon import row_echelon
A = np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
print(row_echelon(A))
```

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

Algoritmo de eliminación gaussiana

Entrada: matriz ampliada $A \in \mathbb{R}^{m\times(n+1)}$.

Salida: matriz escalonada equivalente.

1. Para $i = 1, 2, \ldots, m-1$:
   1. Consideramos la matriz $\tilde{A}$ tomando las filas de A desde la fila i hasta la última.
   2. Buscamos k tal que la columna k de $\tilde{A}$ tenga elementos no nulos, y todas las columnas anteriores sea nulas.
   3. Para $j = i+1, \ldots, m$:
      - Realizamos en A la operación $f_j - \frac{a_{jk}}{a_{ik}} f_i \to f_j$.

Ejercicio: Modificar el algoritmo para contemplar el caso de que el primer elemento de la columna (pivote) sea 0.

Complejidad de eliminación gaussiana

Más sobre complejidad cuando veamos descomposición LU.

Contamos la cantidad de operaciones que necesitamos para escalonar una matriz $A \in \mathbb{R}^{n \times n}$.

- Para la operación de filas $f_j - \frac{a_{jk}}{a_{ik}} f_i \to f_j$ realizamos (a lo sumo) $3n$ operaciones.
- En el paso i, realizamos $n - i$ operaciones de filas.
- En total, en el paso i realizamos $3n(n-i)$ operaciones.
- Sumando sobre todos los $i = 1, 2, \ldots, n-1$, obtenemos

$$3n\left((n-1) + (n-2) + \cdots + (1)\right) = 3n\frac{(n-1)(n)}{2} \approx \frac{3}{2}n^3.$$

La complejidad es de orden n^3 (ignorando las constantes).

Clasificación de sistemas de ecuaciones

Estudiamos ahora cuántas soluciones puede tener un sistema de ecuaciones a partir de la forma escalonada de la matriz ampliada.

Ejemplo

Resolvemos el siguiente sistema de ecuaciones:

$$\begin{cases} 5x_1 + 3x_2 = 11 \\ 15x_1 + 9x_2 = 33 \\ 20x_1 + 12x_2 = 44 \end{cases}$$

Clasificación de sistemas de ecuaciones

Construimos la matriz ampliada

$$\left(\begin{array}{cc|c} 5 & 3 & 11 \\ 15 & 9 & 33 \\ 20 & 12 & 44 \end{array}\right)$$

y escalonamos usando `Python`.

```
A = np.array([[5,3,11],[15,9,33],[20,12,44]])
print(row_echelon(A))
```

Clasificación de sistemas de ecuaciones

```
%% [[1.  0.6 2.2]
%%  [0.  0.  0. ]
%%  [0.  0.  0. ]]
```

Vemos que se eliminaron las últimas dos ecuaciones, y nos queda solo una ecuación:

$$x_1 + 0.6x_2 = 2.2$$

de donde podemos despejar $x_1 = 2.2 - 0.6x_2$.

El sistema tiene infinitas soluciones,

$$S = \{(2.2 - 0.6x_2, x_2) : x_2 \in \mathbb{R}\}.$$

Clasificación de sistemas de ecuaciones

Ejemplo

Consideremos nuevamente el sistema inicial, modificando un valor en $\boldsymbol{b}$:

$$\begin{cases} 5x_1 + 3x_2 = 11 \\ 15x_1 + 9x_2 = 33 \\ 20x_1 + 12x_2 = 55 \end{cases}$$

Clasificación de sistemas de ecuaciones

Escalonamos la matriz ampliada:

```
A = np.array([[5,3,11],[15,9,33],[20,12,55]])
print(row_echelon(A))
```

```
%% [[1.  0.6 2.2]
%%  [0.  0.  1. ]
%%  [0.  0.  0. ]]
```

En la segunda ecuación, obtuvimos $0x_1 + 0x_2 = 1$. ¡Absurdo! El sistema no tiene solución.

Clasificación de sistemas de ecuaciones

Si al escalonar la matriz ampliada, llegamos a una ecuación

$$0x_1 + \cdots + 0x_n = a \neq 0$$

el sistema no tiene solución. Por el contrario, si en todas las filas que se anulan los coeficientes de las variables, obtenemos también 0 en el término independiente, el sistema admite solución (podemos ir resolviendo hacia atrás).

Obtenemos los siguientes casos para un sistema de m ecuaciones y n incógnitas.

Sistema incompatible.

- El sistema no tiene solución. Al escalonar obtuvimos una ecuación $0x_1 + \cdots + 0x_n = a \neq 0$. Ejemplo:

$$\left(\begin{array}{ccc|c} 5 & 3 & 2 & 11 \\ 0 & 9 & -1 & 10 \\ 0 & 0 & 0 & 4 \end{array}\right)$$

Sistema compatible determinado.

- El sistema tiene solución única. Al escalonar la matriz ampliada de un sistema de n incógnitas, obtenemos exactamente n ecuaciones no nulas, y podemos obtener la solución despejando las variables.

$$\left(\begin{array}{ccc|c} 5 & 3 & 2 & 11 \\ 0 & 9 & -1 & 10 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

Sistema compatible indeterminado.

- El sistema tiene infinitas soluciones. Al escalonar la matriz ampliada obtenemos menos de n filas no nulas en las primeras n columnas, y el resto de las filas son nulas en todas las columnas. Ejemplo:

$$\left(\begin{array}{ccc|c} 5 & 3 & 2 & 11 \\ 0 & 9 & -1 & 10 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

Clasificación de sistemas de ecuaciones

Ejercicio

Escalonar las siguientes matrices y clasificar el sistema en (a) incompatible, (b) compatible determinado, (c) compatible indeterminado.

1. $\begin{cases} 3y - 2z + 3w = 9 \\ 2x + y + w = 5 \\ x - y + z - w = -2 \end{cases}$
2. $\begin{cases} x - 2y = 2 \\ 2x + y = 1 \\ x + 3y = -1 \end{cases}$
3. $\begin{cases} 2x + y - z = 3 \\ x - y + z = 2 \\ 5x + y - z = -5 \end{cases}$