Álgebra Lineal Computacional

2do Cuatrimestre 2025

Laboratorio N° 6: Autovalores y autovectores, método de la potencia.

En este laboratorio exploraremos la idea de autovector y autovalor, así como la implementación del Método de la Potencia para encontrar el autovector de autovalor más grande. También aplicaremos reflectores de Householder para calcular todos los autovectores de una matriz simétrica.

Ejercicio 1 (Exploración numérica)**.** La idea básica de autovector es que es una dirección que que se mantiene luego de aplicar una transformación lineal. Dada una matriz $A \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^n$ es autovector de A si

$$Av = \lambda v$$

con $\lambda \in \mathbb{R}$ (en verdad, λ debe pertenecer al cuerpo $\mathbb{K}$ que acompaña al espacio vectorial, pero para este laboratorio no restringimos a los números reales). La forma más sencilla de identificar estas direcciones consiste, esencialmente, en aplicar la matriz A repetidas veces a un vector. Considere la transformación $f_A(v) = w \in \mathbb{R}^n$ definida por las siguientes operaciones:

$$w' = Av$$

$$w = \begin{cases} \frac{w'}{||w'||_2}, & ||w'||_2 > 0 \\ 0, & \textit{en otro caso} \end{cases}$$

Entonces, para un autovector v de A, tenemos que $f_A(v) = signo(\lambda)v$. Llamaremos $f_A^k(v)$ al resultado de aplicar k veces la función f_A a un vector v.

Para los siguientes puntos, considere las matrices:

$$A_1 = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad A_3 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{1}$$

a) Calcule $f_{A_i}^k(v)$ para:

   1. 5 vectores generados al azar,
   2. los siguientes vectores $\left\{\frac{1}{\sqrt{2}}(1,1), \frac{1}{\sqrt{2}}(1,-1), (1,0), (0,1)\right\}$

   Considere $k = 1, 5, 10, 50, 100$, y grafique los vectores obtenidos juntos (un gráfico para cada A_i). ¿En qué casos puede identificar una dirección preferenciada? *Tip: usen colores distintos para cada vector inicial, de forma de poder seguir como se transforman.*

b) El método de la potencia permite encontrar el autovector de mayor autovalor (en módulo) mediante repetidas aplicaciones de la transformación $f_A(v)$. Implemente la función `metpot2k(A,tol=1e-8,K=1000)` que aplique el método de la potencia, según se describe en el Algoritmo 1.

Algoritmo 1: `metpot2k(A,tol=1e-15,K=1000)`, método de la potencia.

Input: Matriz $A \in \mathbb{R}^{n \times n}$, entero K, tolerancia *tol*
Output: Autovector v, autovalor λ, cantidad de operaciones k, error en el último paso ϵ
$v \leftarrow random(\mathbb{R}^n)$
$\tilde{v} \leftarrow f_A^2(v)$
$e \leftarrow \tilde{v}^t v$
$k \leftarrow 0$
while $|e-1| > tol \;\&\; k < K$ **do**
  $v \leftarrow \tilde{v}$
  $\tilde{v} \leftarrow f_A^2(v)$
  $e \leftarrow \tilde{v}^t v$
  $k \leftarrow k+1$
end
$\lambda \leftarrow \tilde{v}^t A \tilde{v}$
$\epsilon \leftarrow e-1$

c) Aplique el método de la potencia a las matrices A_1, A_2 y A_3 y compare con lo obtenido en el punto a).

Ejercicio 2 (Velocidad de convergencia de `metpot2k`)**.** Para estudiar la velocidad de convergencia del Algoritmo 1, realice el siguiente experimento:

- Generar una matriz $C \in \mathbb{R}^{100 \times 100}$ de coordenadas aleatorias y considerar la matriz simétrica $A = \frac{1}{2}(\boldsymbol{C} + \boldsymbol{C}^t)$ (de esta forma nos aseguramos una matriz con todos sus autovalores reales).
- Definir $B = A + 500I$ y aplicar el Algoritmo 1 con un $K = 1000$. Guarde en cada paso del algoritmo la estimación del autovalor al paso k, $\alpha_k = \tilde{v}^t A \tilde{v}$.
- Calcule el error de estimación del autovalor λ en el paso k como $\delta_k = |\lambda - \alpha_k|$. Grafique δ_k en función de k para 10 realizaciones del algorimo (manteniendo la misma matriz C).
- Usando la función `np.linalg.eigvals`, obtenga el autovalor λ_1 de mayor módulo de C y λ_2 el que le sigue. Adicione al gráfico anterior la curva de error estimado teóricamente $\log \delta_k = 2\log(\frac{|\lambda_2|}{|\lambda_1|})k + \delta_0$

Ejercicio 3 (Diagonalización via Householder)**.** Contando con el método de la potencia, el desafío siguiente es como encontrar los otros autovectores y autovalores. Consideremos una matriz $A \in \mathbb{R}^{n \times n}$ simétrica, de forma que sus autovectores sean reales y ortogonales entre sí. Dado un vector $v \in \mathbb{R}^l$, definimos el reflector de Householder $H_v = I_l - 2\frac{(e_1-v)(e_1-v)^t}{||e_1-v||^2}$, donde I_l es la identidad de $l \times l$ y e_1 es el primer canónico de $\mathbb{R}^l$. Se puede mostrar que si la matriz A es simétrica, entonces la matriz $B = H_{v_1} A H_{v_1}^t$, con v_1 el primer autovector de A, está formada por dos bloques:

$$B = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \tilde{A} \end{pmatrix}$$

donde λ_1 es tal que $Av_1 = \lambda_1 v_1$. Gracias a que H_{v_1} es ortogonal, A y B tienen los mismos autovalores, y entonces $\tilde{A}$ tiene los $n-1$ autovalores de A que restan como propios. Podemos combinar esta propiedad con el método de la potencia para encontrar recursivamente todos los autovectores y autovalores de A, como se describe en el Algoritmo 2.

a) Implemente el Algoritmo 2 en la función `diagRH(A,tol=1e-15,K=1e5)`.

b) Apliquelo a las matrices

$$A_1 = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \quad A_2 = \begin{pmatrix} 3/2 & 1/2 & 0 \\ 1/2 & 3/2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad A_3 = \begin{pmatrix} 1 & 3 & 5 \\ 3 & 2 & 7 \\ 5 & 7 & 3 \end{pmatrix} \tag{2}$$

Compare el resultado obtenido con el resultado esperado[1].

[1]Para obtenerlos, realice el cálculo analíticamente o compare en `numpy.linalg.eigvalsh`.

Algoritmo 2: `diagRH(A,tol=1e-15,K=1000)`, Diagonalizacion con MP y RH

Input: Matriz simétrica $A \in \mathbb{R}^{n\times n}$, tolerancia tol, un entero K
Output: Matriz de autovectores S, matriz de autovalores D, tal que $A = SDS^t$
$v_1, \lambda_1 \leftarrow$ `metpot2k(A,tol,K)`
$H_{v_1} \leftarrow I - 2\frac{(e_1-v_1)(e_1-v_1)^t}{||e_1-v_1||^2}$
if n=*2* **then**
  $S \leftarrow H_{v_1}$
  $D \leftarrow H_{v_1} A H_{v_1}^t$
end
else
  $B \leftarrow H_{v_1} A H_{v_1}^t$
  $\tilde{A} \leftarrow B_{2:n,2:n}$
  $\tilde{S}, \tilde{D} \leftarrow$ `diagRH`$(\tilde{A}, tol, K)$
  $D \leftarrow \begin{pmatrix} \lambda_1 & 0 \\ 0 & \tilde{D} \end{pmatrix}$
  $S \leftarrow H_{v_1} \begin{pmatrix} 1 & 0 \\ 0 & \tilde{S} \end{pmatrix}$
end

c) Estudie la calidad de la implementación en función de n mediante dos métricas:

i. Usando la función norma exacta, y los S y D calculados por `diagRH`, compute $||A - SDS^t||_\infty$ en función de n usando matrices simétricas aleatorias.

ii. Usando la función `numpy.linalg.eigvalsh`, compute los autovalores para matrices simetricas aleatorias, y compute $\sum_{i=1}^n |\lambda_i - \hat{\lambda}_i|/n$, donde $\hat{\lambda}_i$ es el valor provisto por `numpy` para el autovalor i-esimo. *Nota: asegurese de comparar los valores en el orden correcto.*

d) Explore la relación entre tol y el error final obtenido. Para esto, fije $n = 10$ y calcule $||A - SDS^t||$ en función de tol para matrices simétricas aleatorias. En todos los casos, asegurese de elegir un K lo suficientemente grande para que el método converja.

Ejercicios extra

Ejercicio 4 (Un poquito de física)**.** El análisis de autovectores se denomina *análisis espectral*, entre varias cosas, por su utilidad para caracterizar frecuencias naturales de sistemas reales. Para tener un *sabor* de esto, vamos a analizar un sistema sencillo: una soga vibrante con extremos fijos. Esto puede representar, por ejemplo, la cuerda de una guitarra. Podemos aproximar la cuerda por un conjunto de n masas, donde el desplazamiento (vertical) de cada una de ellas está descrita por un vector $v \in \mathbb{R}^n$. Sin entrar en demasiado detalle, cualquier movimimiento de la soga puede descomponerse en *modos* (nuestros autovectores) gracias a la existencia de una base de los mismos. Entonces, la evolución de la cuerda en el tiempo queda dada por

$$v(t) = \sum_{i=1}^n \alpha_i v_i \cos(\omega_i t)$$

donde v_i es el autovector de autovalor λ_i, $\omega_i \propto \sqrt{\lambda_i}$ y α_i es la descomposición en la base de la condición inicial, de forma tal que $v(0) = \sum_{i=1}^n \alpha_i v_i$[2]. Los v_i son autovectores resultantes de la matriz que describe la estructura del sistema. Para una cuerda unidimensional:

$$L = \begin{cases} 2, & si\ i = j \\ -1, & si\ i = j \pm 1 \\ 0, & en\ otro\ caso \end{cases}$$

[2]Gracias a que los v_i son ortonormales, $\alpha_i = v_i^t v(0)$.

a) Calcule los autovectores y autovalores de L usando `diagRH` para distintos $n = 5, 10, 50, 100$. Grafique los 5 autovectores con autovalor más bajo para cada n. ¿Puede identificar a qué movimientos de la cuerda corresponden? ¿Qué cambia al aumentar el n?

b) Usando que la evolución de cada *modo* de oscilación es $\omega_i = \sqrt{\lambda_i}$, construya una animación de la evolución temporal de los 5 modos de autovalor más bajo. *Tip: el módulo* `matplotlib.animation` *permite realizar animaciones.*

c) Tomando como condición inicial

$$v(0)_k = \begin{cases} \frac{2k}{n}, & k \leq \frac{n}{2} \\ \frac{2}{n}(n-k), & k > \frac{n}{2} \end{cases}$$

(k refiere al elemento k-esimo del vector $v(0)$), calcule los $\alpha_i = v_i^t v(0)$ y realice una animación del movimiento de la cuerda como $v(t) = \sum_{i=1}^{n} \alpha_i v_i \cos(\omega_i t)$.

d) Repita el punto c) para la condición inicial:

$$v(0)_k = \begin{cases} \frac{10}{n}k, & k \leq \frac{n}{10} \\ \frac{10}{n}(\frac{n}{5}-k), & \frac{n}{10} < k \leq \frac{n}{5} \\ 0, & k > \frac{n}{5} \end{cases}$$

Módulo ALC

Para el módulo ALC, deben programar las siguientes funciones. Tests para las mismas acompañan en el archivo `L05.py`.

```python
def metpot2k(A, tol=1e-15,K=1000):
    """
    A una matriz de n x n
    tol la tolerancia en la diferencia entre un paso y el siguiente de la estimacion del
     autovector.
    K el numero maximo de iteraciones a realizarse.
    Retorna vector v, autovalor lambda y numero de iteracion realizadas k
    """

def diagRH(A, tol=1e-15,K=1000):
    """
    A una matriz simetrica de n x n
    tol la tolerancia en la diferencia entre un paso y el siguiente de la estimacion del
     autovector.    K el numero maximo de iteraciones a realizarse
    retorna matriz de autovectores S y matriz de autovalores D, tal que A = S D S.T
    Si la matriz A no es simetrica, debe retornar None
    """
```

Nota: no está permitodo el uso de la multiplicación matricial de numpy (`@`, `np.matmul`, etc)