Álgebra Lineal Computacional

2do Cuatrimestre 2025

Laboratorio N° 03: Normas y condición de matrices.

1. Normas

Sea $x \in \mathbb{R}^n$. Se define la norma $p \in \mathbb{N}$ de x como:

$$||x||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

Ejercicio 1 (Puntos de norma 1)**.** Desarrolle:

a) Una función `norma(x,p)` que reciba un vector x (un objeto iterable) y una norma p y retorne su norma.

b) Una función `normaliza(X,p)` que reciba una lista de vectores `X` y una norma p y retorne una lista `Y` donde cada elemento corresponde a normalizar los elementos de `X` con la norma p.

Emplee la función construida para graficar, en un mismo plot, los vectores con norma 1 de $\mathbb{R}^2$ según las normas $p = 1, 2, 5, 10, 100, 200$. Describa las figuras que observa. Identifique a qué forma funcional converge la norma para valores de $p \longrightarrow +\infty$. Muestre gráficamente que ese límite puede expresarse con la norma infinito y

c) Modifique la función `norma(x,p)` para que pueda recibir el argumento `p='inf'` con

$$||x||_\infty = \max_{i=1,\ldots,n} |x_i|$$

d) Incorpore la norma infinito al gráfico anterior.

e) ¿En qué valores coinciden todas las normas graficadas?

2. Normas matriciales inducidas

La norma q, p inducida de una matriz $A \in \mathbb{R}^{m \times n}$ se define como

$$||A||_{q,p} = \max_{x/||x||_p=1} ||Ax||_q$$

para dos normas $||.||_p : \mathbb{R}^n \longrightarrow \mathbb{R}^+$ y $||.||_q : \mathbb{R}^m \longrightarrow \mathbb{R}^+$. Noten que las normas no tienen por qué ser idénticas, ni la matriz A necesita ser cuadrada.

Ejercicio 2 (Normas q, p)**.** En este ejercicio desarrollaremos funciones aleatorias y exactas para calcular las normas inducidas de la matriz.

a) Aplicamos el método de *Monte Carlo* para estimar la norma inducida de una matriz. Para ello, implementar una función `normaMatMC(A,q,p,Np)` que estime $||A||_{q,p}$ numéricamente, usando `Np` vectores x de $\mathbb{R}^n$ generados al azar. La función debe retornar la norma $||A||_{q,p}$ y el vector x en el cual se alcanza el máximo.

b) Use la función construida para estimar las siguientes normas, junto a las posiciones en las que se alcanzan:

i. $||I||_{2,1}$, $I \in \mathbb{R}^{2,2}$

ii. $||I||_{1,2}$, $I \in \mathbb{R}^{2,2}$

iii. $||I||_{2,\infty}$, $I \in \mathbb{R}^{2,2}$

iv. $||I||_{\infty,2}$, $I \in \mathbb{R}^{2,2}$

v. $||A||_{2,2}$, $A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$

vi. $||A||_{2,2}$, $A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

vii. $||A||_{\infty,\infty}$, $A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$

viii. $||A||_{2,\infty}$, $A = \begin{pmatrix} 10 & 10 \\ 0 & 0 \end{pmatrix}$

donde I es la matriz identidad.

En todos los casos, calcule 100 veces la norma, usando `Np=1000`, de forma de generar una muestra de la misma. ¿Siempre ese estable el valor que obtiene? ¿Por qué? Inteprete geométricamente los resultados.

c) Construya una función `normaExacta(A,p=[1,'inf'])` que calcule las normas 1 e ∞ de una matriz A usando las expresiones:

$$||A||_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |A_{ij}| \qquad ||A||_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |A_{ij}|$$

y comparelas con lo obtenido en el punto anterior. *Nota: cuando las normas de partida y llegada son las mismas ($p = q$) la norma $||A||_{p,p}$ de nota simplemente $||A||_p$*

3. Condicionamiento de matrices

Una matriz $A \in \mathbb{R}^{n\times n}$ se dice bien condicionada si al resolver el problema

$$Ax = b$$

pequeñas variaciones en el valor de b no producen grandes variaciones en la solución x que encontramos. En este contexto, resulta útil el número de condición

$$\kappa_p(A) = ||A||_p||A^{-1}||_p \tag{1}$$

que permite dar una cota a cómo se propaga el error cometido debido a una pequeña variación en el vector de soluciones:

$$\frac{||x-\tilde{x}||_p}{||x||_p} \leq \kappa_p(A)\frac{||b-\tilde{b}||_p}{||b||_p}$$

donde $Ax = b$ y $A\tilde{x} = \tilde{b}$.

Ejercicio 3 (Numero de condición)**.** Desarrolle las funciones:

a) `condMC(A,p)` que estime el número de condición de A usando la norma inducida p. La función debe emplear la función `normaMatMC` desarrollada anteriormente[1].

b) Desarrolle una rutina para poner a prueba la cota dada por el número de condición. Para esto:

   i. Desarrolle una funcion `variaPerc(b,perc)` que reciba un vector $b \in \mathbb{R}^n$ y varíe sus elementos en un porcentaje aleatorio. Identifique qué norma es más sensible a esta variación. *Tip:* Una forma de realizar esto es multiplicar cada elemento de b por un número aleatorio entre $1 - perc/100$ y $1 + perc/100$. De esta forma, tendremos que $||b - \tilde{b}||/||b|| \leq perc/100$.

   ii. Construya una función que reciba a una matriz `A`, una norma `p`, un porcentaje `perc` y una cantidad de realizaciones `Np` y siga el siguiente esquema para encontrar la peor combinación de b y $\tilde{b}$ para la matriz A:

      - genera `Np` instancias de b y `Np` instancias de $\tilde{b}$, donde la variación porcentual es a lo sumo `perc`.
      - resuelve $Ax = b$ y $A\tilde{x} = \tilde{b}$.
      - Calcula $e_x = ||x - \tilde{x}||/||x||$ y $e_b = ||b - \tilde{b}||/||b||$
      - Calcula el número de condición de A.
      - Retorna en una lista el número de condición, los 2`Np` errores de e_x y e_b, y los valores de b y $\tilde{b}$ en los que se alcanza el máximo de e_x

   Empleando esta función, grafiquen un histograma de $\frac{||x-\tilde{x}||\cdot||b||}{||x||\cdot||b-\tilde{b}||}$ marcando con una línea vertical el número de condición de cada una de las matrices A presentadas a continuación, para las normas 1, 2 e infinito, y porcentages `perc=1,5,10`. ¿Cuán cercanos son estos valores al número de condición?

      - $A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$
      - $A = \begin{pmatrix} 1000 & 1/1000 \\ 0 & 1000 \end{pmatrix}$
      - $A = \begin{pmatrix} 501 & 499 \\ 500 & 500 \end{pmatrix}$
      - $A = \begin{pmatrix} 1/1000 & 1000 \\ 0 & 1000 \end{pmatrix}$

   iii. Usando la función del punto anterior, realice 10 búsquedas de vectores b y $\tilde{b}$ en los cuales el error $||x - \tilde{x}||/||x||$ es máximo para cada una de las normas y porcentajes de error considerados. Use `Np=100000` muestras para cada una de las matrices listadas en el punto anterior. Compute $\Delta b = \tilde{b} - b$ y grafíque b y Δb usando la función `arrow` de `matplotlib`, empleando una flecha para mostrar b y otra que indique Δb partiendo desde b. ¿Cuán estables son los vectores obtenidos para cada matriz y porcentaje? Compare como varía el resultado obtenido al aumentar `perc` y la definición de norma. ¿Varía la orientación de Δb con `perc`?

[1] *Nota: en el contexto de esta práctica, aún no disponemos de una función* `inversa(A)` *propia. Por esto, pueden emplear* `numpy.linalg.inv` *momentáneamente para el cálculo de la inversa. Al entregar el módulo ALC deben reemplazar estas instancias por las funciones desarrolladas en el curso.*

Ejercicios extra

Ejercicio 4 (Optimización por Luus-Jaakola)**.** A lo largo de este labo, deberían haberse preguntado sobre los tiempos de corrida para los análisis realizado. En particular, probar en `Np` valores al azar es muy ineficiente: si queremos encontrar un mínimo o máximo dado que las funciones son continuas debería haber una estrategia mejor. Si bien más adelante vamos a programar metodologías más directas, un pequeño paso a esta altura nos puede ahorrar muchos cálculos.

Programen un buscador de Luus-Jaakola para mejorar la función del punto 2.a `normaMatLJ(A,q,p,Np,maxiter=1000,variacion=2,rate=0.95)`. Se agregan los parámetros `maxiter`, que limita el número máximo de iteraciones del algoritmo y `variacion` que indica la variabilidad de los puntos generados. El algoritmo se inicia generando `Np` vectores de norma unitaria de acuerdo a la norma `p` y un contador `iter`. Tomamos como x_0 al elemento de esta primer muestra tal que $||Ax_0||_q$ es máximo. Luego repetimos hasta llegar al máximo de iteraciones:

- Generamos `Np` muestras centradas en x_0 con valores uniformemente distribuidos entre `x0 + variacion` y `x0-variacion`. (`np.random.uniform`).
- Se normalizan los nuevos vectores generados usando la norma p.
- Basandonos en la muestra de vectores normalizados:
  - Si existe un x_1 en la nueva muestra tal que $||Ax_1||_q > ||Ax_0||_q$, entonces reemplazamos $x_1 = x_0$
  - Si se mantiene que $||Ax_0||_q$ es el máximo encontrado, hacemos `variacion = variacion*rate`
- Avanzamos un paso `iter += 1`.

Módulo ALC

Para el módulo ALC, deben programar:

```python
def norma(x,p):
"""
Devuelve la norma p del vector x.
"""

def normaliza(X, p):
"""
Recibe X, una lista de vectores no vacios, y un escalar p. Devuelve
    una lista donde cada elemento corresponde a normalizar los
    elementos de X con la norma p.
"""

def normaMatMC(A,q,p,Np):
"""
Devuelve la norma ||A||\_{q,p} y el vector x en el cual se alcanza
    el maximo.
"""

def normaExacta(A,p=[1,'inf']):
"""
Devuelve una lista con las normas 1 e infinito de una matriz A
    usando las expresiones del enunciado 2.(c).
"""

def condMC(A, p):
"""
Devuelve el numero de condicion de A usando la norma inducida p.
"""

def condExacto(A, p):
"""
Que devuelve el numero de condicion de A a partir de la formula de
    la ecuacion (1) usando la norma p.
"""
```

Los tests acompañan en un archivo `.py`.