OPTI- Lab 2

Marcelino Sánchez Rodríguez 191654

2024-02-14

Definimos función

```
import numpy as np

def mi_esp_nulo(Q, A, c, b) :

    #-------------------
    # M´etodo del espacio nulo para el problema cuadr´atico convexo
    # Min (1/2) * x.T * Q * x + c.T * x
    # s. a A * x = b
    #-------------------
    (m, n) = np.shape(A)
    #-------------------
    # Descomposici´on en valores singulares
    (U, S, Vh) = np.linalg.svd(A, full_matrices = True)
    V = Vh.T
    V1 = V[:, 0 : m]
    #--Base del espacio nulo---------
    Z = V [:, m : n]
    #-------------------
    # Soluci´on Particular / A * xpar = b
    xpar = np.dot(U.T, b)
    Sinv = 1/S
    xpar = Sinv * xpar
    xpar = np.dot(V1, xpar)
    #-------------------
    # matriz del problema cuadr´atico convexo sin restricciones
    QZ = np.dot(Z.T, Q)
    QZ = np.dot(QZ, Z)
    #-------------------
    # Lado derecho
    ld = np.dot(Q, xpar) + c
    ld = -np.dot(Z.T, ld)
    #---soluci´on del problema cuadr´atico sin restricciones
    xz = np.linalg.solve(QZ, ld)
    #--------------------
    # Soluci´on del problema original
    xstar = xpar + np.dot(Z, xz)
    return xstar
```

Ejemplo

```
import numpy as np
m = 5
n = 9
A = np.random.randn(5, 9)
b = np.ones(m)
c = 10 * np.random.rand(n)
vd = np.arange(1, n + 1)
Q = np.diag(vd)
#
xstar = mi_esp_nulo(Q, A, c, b)
#
print("Soluci´on del problema cuadr´atico ---")
for i in range(len(xstar)) :
    print(f"x[{i}] = ", xstar[i])
```

```
Soluci´on del problema cuadr´atico ---
x[0] =  -0.5798121379763456
x[1] =  -1.1266018431130864
x[2] =  -0.10197793532133924
x[3] =  -0.8619682566086834
x[4] =  -0.6938307827580608
x[5] =  0.5347542491598314
x[6] =  0.3975458763412308
x[7] =  0.7443976992018659
x[8] =  0.03375182467984161
```

```
K= np.concatenate((Q,A.T),1)
MC = np.zeros((m,m))
K1= np.concatenate((A,MC),1)
K2= np.concatenate((K,K1),0)

f = np.concatenate((-c,b),0)

w = np.linalg.solve(K2,f)

ystar = w[0:n]

for i in range(len(xstar)) :
    print(f"y[{i}] = ", ystar[i])
```

```
y[0] =  -0.5798121379763466
y[1] =  -1.1266018431130875
y[2] =  -0.10197793532133975
y[3] =  -0.8619682566086838
y[4] =  -0.6938307827580606
y[5] =  0.5347542491598323
y[6] =  0.3975458763412313
y[7] =  0.7443976992018669
y[8] =  0.03375182467984188
```

```
verror = xstar - ystar
error = np.linalg.norm(verror)

print("Norma del Error = ", error)
```

```
Norma del Error =  2.193305226157645e-15
```