Sparse Matrices

November 2, 2020

0.0.1 EJEMPLO 1

Podemos pasar de una matriz densa a una dispersa usando la función `csr_matrix()` y esta nueva matriz regresarla a una matriz densa con la función `todense()`.

[2]:
```
# Matriz dispersa

from numpy import array
from scipy.sparse import csr_matrix
# Creando una matriz densa
A = array([
    [1, 0, 0, 1, 0, 0],
    [0, 0, 2, 0, 0, 1],
    [0, 0, 0, 2, 0, 0]
])
print(A)
# Convirtiendo a una matriz dispersa (Método CSR)
S = csr_matrix(A)
print(S)
# Reconstruyendo la matriz densa
B = S.todense()
print(B)
```

```
[[1 0 0 1 0 0]
 [0 0 2 0 0 1]
 [0 0 0 2 0 0]]
  (0, 0)        1
  (0, 3)        1
  (1, 2)        2
  (1, 5)        1
  (2, 3)        2
[[1 0 0 1 0 0]
 [0 0 2 0 0 1]
 [0 0 0 2 0 0]]
```

0.0.2 EJEMPLO 2

Usando la función `count_nonzero()` podemos obtener el número de elementos diferentes de cero y así calulcar la escasez tal como indica la fórmula descrita anteriormente.

[9]:
```
# Escasez de una matriz

from numpy import array, count_nonzero
# Creando una matriz densa
A = array([
    [1, 0, 0, 1, 0, 0],
    [0, 0, 2, 0, 0, 1],
    [0, 0, 0, 2, 0, 0]
])
print(A)
# Calculando la escasez
escasez = 1 - (count_nonzero(A)/ A.size)
print(escasez)
```

```
[[1 0 0 1 0 0]
 [0 0 2 0 0 1]
 [0 0 0 2 0 0]]
0.7222222222222222
```