wikipedia_wisconsin_starter

April 11, 2022

[8]:
```
import numpy as np
from scipy.sparse import csc_matrix
from scipy.sparse.linalg import eigs

edges_file = open('wisconsin_edges.csv', "r")
nodes_file = open('wisconsin_nodes.csv', "r")

# create a dictionary where nodes_dict[i] = name of wikipedia page
nodes_dict = {}
for line in nodes_file:
    nodes_dict[int(line.split(',',1)[0].strip())] = line.split(',',1)[1].strip()

node_count = len(nodes_dict)

# create adjacency matrix
A = np.zeros((node_count, node_count))
for line in edges_file:
    from_node = int(line.split(',')[0].strip())
    to_node = int(line.split(',')[1].strip())
    A[to_node, from_node] = 1.0

## Add code below to (1) prevent traps and (2) find the most important pages
# Hint -- instead of computing the entire eigen-decomposition of a matrix X using
# s, E = np.linalg.eig(A)
# you can compute just the first eigenvector with:
# s, E = eigs(csc_matrix(A), k = 1)
```

[9]:
```
for i in range(len(A)):
    for j in range(len(A)):
        A[i][j] += 0.001
A
```

[9]:
```
array([[0.001, 0.001, 0.001, …, 0.001, 0.001, 0.001],
       [0.001, 0.001, 0.001, …, 0.001, 0.001, 0.001],
       [0.001, 0.001, 0.001, …, 0.001, 0.001, 0.001],
```

```
       …,
       [0.001, 0.001, 0.001, …, 0.001, 0.001, 0.001],
       [0.001, 0.001, 0.001, …, 0.001, 0.001, 0.001],
       [0.001, 0.001, 0.001, …, 0.001, 0.001, 0.001]])
```

[13]:
```
A_norm = A/A.sum(axis=0, keepdims=1)
```

[15]:
```
s,E = eigs(csc_matrix(A_norm),k=1)
s,E
```

[15]:
```
(array([1.+0.j]),
 array([[-0.00094793+0.j],
        [-0.00113526+0.j],
        [-0.00094793+0.j],
        …,
        [-0.01864669+0.j],
        [-0.00164875+0.j],
        [-0.00094793+0.j]]))
```

[16]:
```
E = abs(E)
```

[20]:
```
e_dict = {}

for i in range(len(E)):
    e_dict[i] = E[i]

e_dict_sorted = sorted(e_dict.items(), key=lambda x : x[1], reverse=True)
print("B: " , e_dict_sorted[0][0])
print("C: ", e_dict_sorted[2][0])
```

```
B:  5089
C:  1345
```