東京工業大学
Tokyo Institute of Technology

Complex Network

Quiz 4

WANG BIYUAN

18M38156

Department of Computer Science

Lecturer: Tsuyoshi Murata

December 14, 2018

Code

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np

G = nx.Graph()
G.add_nodes_from(range(0,10))
G.add_edges_from([(0,2),(1,2),(2,3),(2,4),(3,5),(3,6),(4,6),(5,7),\
                  (6,7),(7,8),(8,9),(8,10)])
plt.figure(figsize=(5, 5))
nx.draw_spring(G, node_size=400, node_color='red', with_labels=True,
    font_weight='bold')
if nx.is_connected(G):
  A = nx.adjacency_matrix(G).todense()
  A = np.array(A, dtype = np.float64)
  c = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]) # counter
  walkLength = 50000
  id = 0
  visited = [id]
  c[id] = c[id] + 1
  for k in range(walkLength):
    p = A[id,:]
    id = np.random.choice(np.flatnonzero(p == p.max()))
    visited.append(id)
    c[id] = c[id] + 1
  print(c)
else:
  print("not connected!")
```

Results

What happens when a walker keeps walking for a long time on a graph?
Every vertex in a path is accessible from every other vertex since the graph G is connected and it is a finite state chain, therefore, it follows that it has a unique invariant distribution, which we shall denote π. If a walker keeps walking for a long time tends to infinity, p(t) converges to π.

1. graph 1: find AD^{-1}
AD^{-1} of graph 1 is calculated as follows:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$k_1 = \sum_{i=1} A_{ij} = 0 + 1 + 0 + 0 = 1$

$k_2 = \sum_{i=2} A_{ij} = 1 + 0 + 1 + 1 = 3$

$k_3 = \sum_{i=3} A_{ij} = 0 + 1 + 0 + 1 = 2$

$k_3 = \sum_{i=4} A_{ij} = 0 + 1 + 1 + 0 = 2$

$$D = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \text{ therefore, } D^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}.$$

$$AD^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{3} & 0 & 0 \\ 1 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{3} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{2} & 0 \end{bmatrix}$$

2. graph 1: find $p_0(\infty), p_1(\infty), p_2(\infty)$, and $p_3(\infty)$
When time tends to infinity, $p(t) = AD^{-1}p(t-1) \rightarrow p = AD^{-1}p$

$$\begin{aligned} p &= AD^{-1}p \\ &= (I - AD^{-1})p \\ &= (D - A)D^{-1}p \\ &= LD^{-1}p =, \end{aligned} \tag{1}$$

where $D^{-1}p$ is an eigenvector of the Laplacian with eigenvalue 0. In a connected network, only one eigenvector (with eigenvalue 0) whose components are all equal, therefore, we can obtain an equation according to the graph connection:

$$\begin{aligned} p_i &= \frac{k_i}{\sum_j k_j} = \frac{k_i}{2m} \\ x(\text{node } 0) + 3x(\text{node } 1) + 2x(\text{node } 2) + 2x(\text{node } 3) &= 1 \\ x &= \frac{1}{8} \\ p_0(\infty) &= \frac{1}{8} \\ p_1(\infty) &= \frac{3}{8} \\ p_2(\infty) &= \frac{2}{8} \\ p_3(\infty) &= \frac{2}{8} \end{aligned} \tag{2}$$

3. graph 2: make a program of simulating random walks. Show its results and answer the most visited node.

The number of visit to each node is [2169 2214 8619 6284 4256 4134 6331 6162 5916 1910 2006]. The most visited node is Node 2 (8619 times) because it has the highest degree.