Introduction to Network Analysis 2022/23 (iv)

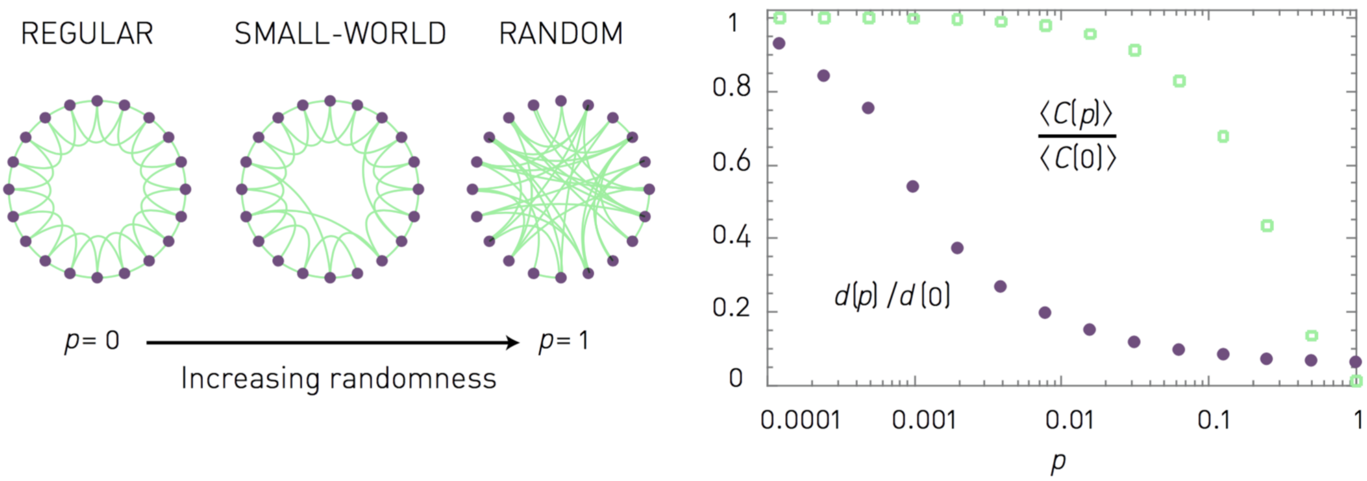
Small-world and scale-free models, graphs vs networks

You are given six networks in Pajek format.

- Zachary karate club network (karate_club.net)
- Map of Darknet from Tor network (darknet.net)
- iMDB actors collaboration network (collaboration_imdb.net)
- WikiLeaks cable reference network (wikileaks.net)
- Enron e-mail communication network (enron.net)
- A small part of Google web graph (www_google.net)

I. Watts-Strogatz small-world graphs

1. **(discuss)** Study the algorithm for generating Watts-Strogatz small-world graphs $G(n, k, p)$ introduced in lectures. Does the algorithm generate networks with realistic structure? What is the time complexity of the algorithm?



1. **(code)** Implement the algorithm and generate Watts-Strogatz small-world graphs that best match the networks above. *(Set k to $\langle k\rangle$ rounded to the nearest even number and try to find the value of p that best reproduces $\langle C\rangle$, $p \approx 1-\sqrt[3]{\frac{4C(k-1)}{3(k-2)}}$.)* Compute their average node clustering coefficient $\langle C\rangle$ and approximate average distance between the nodes $\approx \langle d\rangle$. Are the results expected?

II. Barabási-Albert and Price scale-free graphs

1. **(discuss)** Study the following two algorithms for generating Barabási-Albert scale-free graphs $G(n, c)$

and Price scale-free graphs $G(n, c, a)$ using the relation $\frac{q+a}{n(c+a)} = \frac{c}{c+a}\frac{q}{nc} + \frac{a}{c+a}\frac{1}{n}$. What is the main difference between the algorithms? What is the time complexity of the algorithms?

```
input   nodes n, degree c
output  undirected scale-free G
Q ← empty queue
G ← empty graph
while not G has n nodes do
    i ← add node to G
    for c times do
        Q.add(i)
        Q.add(j ← Q.random())
        add link between i and j
return G
```

```
input   nodes n, out-degree c, free a
output  directed scale-free G
Q ← empty queue
G ← c isolated nodes
while not G has n nodes do
    i ← add node to G
    for c times do
        if [0, 1).random() < c/(c + a) then
            Q.add(j ← Q.random())
        else
            Q.add(j ← {0, . . . , i}.random())
        add link from i to j
return G
```

2. **(code)** Implement both algorithms and generate Barabási-Albert and Price scale-free graphs corresponding to larger networks above. Plot their degree distribution p_k and compute power-law exponents γ of seemingly scale-free distributions using the maximum likelihood formula below. Are the results expected?

$$\gamma = 1 + \bar{n}\left[\sum_{i=1}^{n} \ln \frac{k_i}{k_{min} - \frac{1}{2}} \delta(k_i \geq k_{min})\right]^{-1}$$

III. Synthetic random graphs vs real networks

Consider different large-scale properties of real networks. Namely, low average node degree $\langle k \rangle \ll n$, one giant connected component $S \approx 1$, short distances between the nodes $\langle d \rangle \approx \frac{\ln n}{\ln \langle k \rangle}$, high average node clustering coefficient $\langle C \rangle \gg 0$, power-law degree distribution $p_k \sim k^{-\gamma}$, pronounced community structure etc.

1. **(discuss)** Design synthetic graph model that generates undirected graphs that are *most different* from real networks.

2. **(code)** Implement generative graph model that *well reproduces* the structure of real undirected networks.

3. **(discuss)** Does your model have reasonable interpretation or explanation? Does it also reproduce the structure of real directed networks?