

# Network Complexity

Thesis Subtitle

Yipei Zhao

A dissertation presented for the degree of  
Master of Science



MSc Data Analytics

Aston University

United Kingdom

05 September 2021

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Random graphs . . . . .	2
1.2	Rewiring . . . . .	2
1.3	Small-world . . . . .	2
1.4	Scale-free network . . . . .	3
<b>2</b>	<b>Methods</b>	<b>4</b>
2.1	Implemeted methods . . . . .	4
2.2	$MA_{RI}$ . . . . .	5
2.3	Potential problems and solutions of different subgraph measures . . .	7
<b>3</b>	<b>Result</b>	<b>9</b>
3.1	Complexity measures on small graphs . . . . .	9
<b>4</b>	<b>Conclusion</b>	<b>10</b>

# 1 Introduction

In my literature review, several complexity measures were introduced, includes the theory and the difference between them.

## 1.1 Random graphs

We have many real networks in the actual world, but defining or observing all of them is not feasible. For simulations and comparisons, network scientists introduced the idea of random networks. They are also known as Erdos-Renyi network in honour of two mathematicians: Paul Erdos and Alfred Renyi. They have important contributions to understand the properties of a random network[2].

There are two definitions of a random network:

- $G(n, p)$  network. A network with  $n$  nodes will be initialised, there will be at most  $(n)(n - 1)/2$  edges. Each edge will be instantiated with probability  $p$ . This approach brings a randomness property to the graph; number of edges  $m$ . A  $G(n, p)$  graph returns a fixed  $n$  but a different  $m$  everytime. The expectation of  $m$  is equal to  $p(n)(n - 1)/2$ .
- $G(n, m)$  or  $G(n, L)$  network. A graph with  $n$  nodes will be initialised,  $m/L$  edges will be connected from a random node to another random node. Due to the non-randomness of  $G(n, m)$  networks, they are used to simulate the behaviour of a random network in this thesis.

In the literature review, we introduced the idea of clustering coefficient and average distance. For a random graph, the clustering coefficient and average distance can be calculated using formulas.[2] The average clustering coefficient of a random graph is  $p$ , or  $2m/((n)(n - 1))$  (number of instantiated links divided by total number of possible links). Clustering coefficient is used to illustrate the ratio between connected links and possible links between a node's neighbours. If there are  $k$  neighbours of a node, there can be at most  $k(k - 1)/2$  between the neighbours. In these  $k(k - 1)/2$  links, only  $p$  of them will be instantiated. Thus, the ratio of connected links and possible links becomes  $\frac{pk(k-1)/2}{k(k-1)/2} = p$ . Additionally, average distance of a random graph is  $\langle d \rangle \approx \frac{\ln(n)}{\ln(k)} \approx \frac{\ln(n)}{\ln(2m/n)}$ . To be noticed, both parameters are expectation/approximated, they won't be exact for a random graph.

## 1.2 Rewiring

## 1.3 Small-world

About 50 years ago, a famous study was carried out by Stanley Milgram[5] in the interest of this question: how many intermediates are needed to pass a message between two irrelevant or distanced person? This is known as the small-world problem. As

counterintuitive as it may seem, the medium number of intermediates needed is only 5 (an average of 6). This is not a fair and undoubtable experiment and it is almost impossible to determine the actual number of intermediates needed in modern world. Nevertheless, this number would be smaller than most people's expectation. Mathematically, the small world problem is the study of graphs with small path length. Previously, we introduced the formula to calculate the average distance  $d_r$  of a random graph. Thus, if a graph has  $\langle d \rangle / d_r < 1$ , this graph has less average distance than random graphs. If the ratio  $d / d_r$  is relatively small, we can classify it as a small-world network.

A small-world network can be generated by a Watts-Strogatz (WS) model [7] or a Newman-Watts (NW) model (a variant of the WS model) [6]. Both models require three parameters: number of nodes  $n$ , number of connected closest neighbours  $k$  and rewiring probability  $p$ . The key of the model is rewiring. The graph starts with  $n$  nodes, each node is connected to  $k$  ( $k - 1$  if  $k$  is odd) nearest neighbours;  $nk/2$  edges will be created. For each edge  $(u, v)$ , there is a probability  $p$  that a new edge  $(u, w)$  is created (maintains the starting node but connects to another random node). While rewiring, the WS model removes the edge  $(u, v)$ , thus, the number of edges stays the same. However, the NW model maintains the edge  $(u, v)$ , causing the expectation of number of edges after rewiring to be  $nk/2 + pnk/2$ . Rewiring will add short path to the networks, and cause the average distance to be exceptionally smaller. Suggested by Barabasi [2], to obtain both high clustering and low average distance,  $p$  should be between 0.001 and 0.1.

## 1.4 Scale-free network

A controversial topic of network science is whether real networks are usually scale-free. To state the definition of scale-free, we need to scope into the degree distribution of graphs.

Suggested by Barabasi [2], the degree distribution of a random graph is expected to follow a Poisson distribution. However, Poisson distribution is not the ideal distribution of a real network. A controversial idea that hasn't yet been proven in network science community is: are real networks' degree distribution follows a power-law distribution? [3] A power-law distribution follows:  $P(k) \sim k^{-\gamma}$ , the parameter  $\gamma$  is typically in the range  $2 < \gamma < 3$ . If the degree distribution of a graph follows power-law distribution, the graph is said to be a scale-free network. Even though there are counter-examples, many network scientists still believe that real networks are scale-free. In order to further simulate the behaviour of real networks, Barabasi introduced the Barabasi-Albert (BA) model to create scale-free networks. [2].

The BA model requires two parameters:  $n$  and  $m$ . Initially, only one node is created. Whenever a node is added into the network, it will connect to  $m$  nodes. The logic of connection is the key of this model. Nodes are more likely to connect to nodes

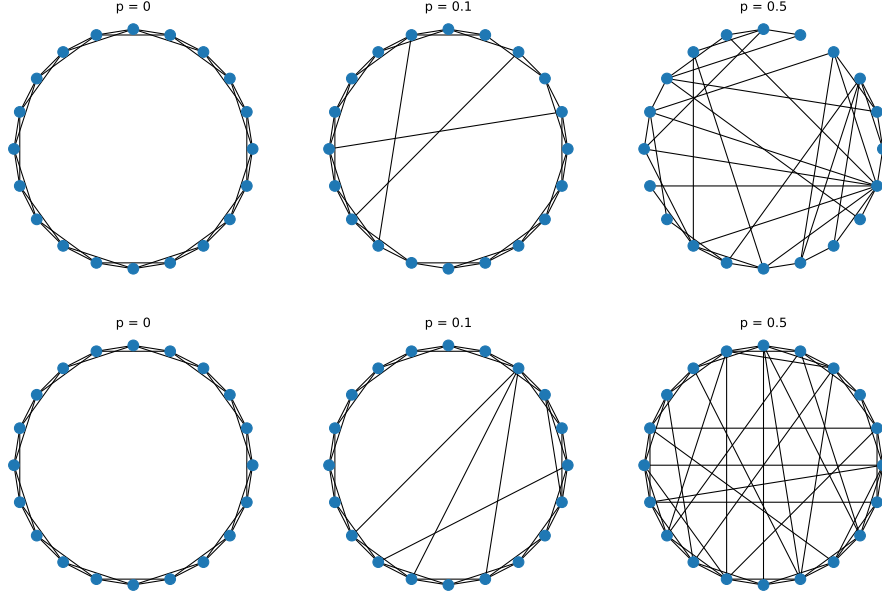


Figure 1: A demonstration of WS model and NW model. The parameters are:  $n = 20$ ,  $k = 4$ ,  $p = 0, 0.1, 0.5$ .

with more links than nodes with less links. For instance, a node has been added to the network, it is more likely to connect to a node with 7 links than a node with 3 links. This logic of connection is called preferential attachment. Essentially, like in real world, nodes are more likely to connect to another node that has more impact on the network. The BA model ensures most of the nodes have low degree, whereas only a few nodes have exceptionally high degree, as shown in figure 2. An ideal way to fit the power-law distribution is using a linear regression to fit the data in log-log scale.

## 2 Methods

### 2.1 Implemented methods

In the literature review, 9 methods were introduced, 7 methods were successfully implemented and tested with a new method *MAri* based on the idea of *MAg*. The implemented methods are:

- Subgraph measures:

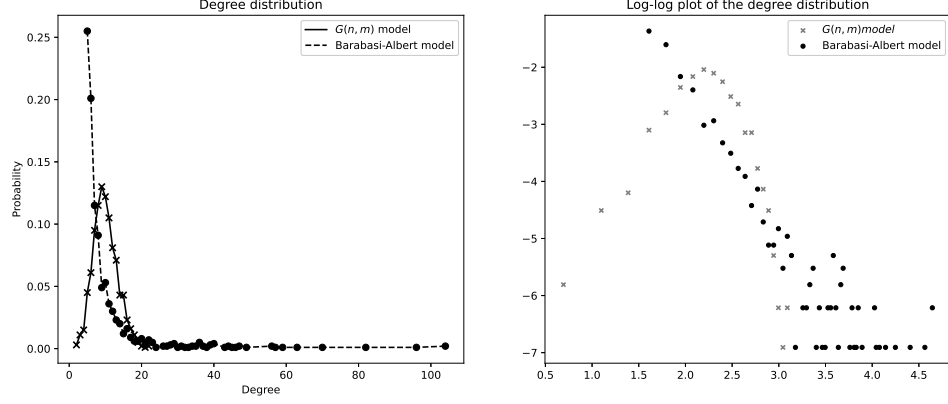


Figure 2: Degree distribution of a  $G(n, m)$  graph with  $n = 1000, m = 5000$  and a graph generated by Barabasi-Albert model with  $n = 1000, m = 5$ .

- $C_{1e,st}$
- $C_{1e,spec}$
- $C_{2e,spec}$
- $OdC$  (Entropy measure)
- Product measures:
  - $MAg$
  - $Cr$
  - $Ce$
  - $MAri$

## 2.2 $MA_{RI}$

The  $MAg$  measure is a product measure, which distributes higher complexity to graphs with medium number of edges and lower complexity at both tails. Using the product of redundancy  $R$  and mutual information  $I$ , with normalisation,  $MAg$  is defined as[4]:

$$\begin{aligned}
 R &= \frac{1}{m} \sum_{i,j>i} \ln(d_i d_j) \\
 I &= \frac{1}{m} \sum_{i,j>i} \ln\left(\frac{2m}{d_i d_j}\right)
 \end{aligned} \tag{1}$$

$$\begin{aligned}
MA_R &= 4\left(\frac{R - R_{path}}{R_{clique} - R_{path}}\right)\left(1 - \frac{R - R_{path}}{R_{clique} - R_{path}}\right) \\
MA_I &= 4\left(\frac{I - I_{clique}}{I_{path} - I_{clique}}\right)\left(1 - \frac{I - I_{clique}}{I_{path} - I_{clique}}\right) \\
MA_g &= MA_R * MA_I
\end{aligned} \tag{2}$$

$I$  can be written as:

$$\begin{aligned}
I &= \frac{1}{m} \sum_{i,j>i} \ln\left(\frac{2m}{d_i d_j}\right) \\
I &= \frac{1}{m} \left( \sum_{i,j>i} \ln(2m) - \sum_{i,j>i} \ln(d_i d_j) \right) \\
I &= \frac{1}{m} \sum_{i,j>i} \ln(2m) - \frac{1}{m} \sum_{i,j>i} \ln(d_i d_j) \\
I &= \ln(2m) - R
\end{aligned} \tag{3}$$

$R_{path}$ ,  $R_{clique}$ ,  $I_{path}$  and  $I_{clique}$  represent the lowest redundancy, highest redundancy, highest mutual information and lowest mutual information of graphs with fixed  $m$  and  $n$  respectively. The equations can be found in the literature review. Kim and Wilhelm suggested that network scientists may use  $C = (R - R_{path})(I - I_{clique})$  as a complexity measure, however, the upperbound cannot be found to normalise the complexity. From our study, an upper-bound of  $C$  can be calculated analytically. Assuming the upper-bound  $C_{max}$  can be found,  $0 < C/C_{max} < 1$ . As suggested in equation set 3,  $I = \ln(2m) - R$ , we can rewrite the complexity equation:

$$C = (R - R_{path})(\ln(2m) - R - I_{clique}) \tag{4}$$

$$C = -R^2 + (\ln(2m) - I_{clique} + R_{path})R + (-R_{path}\ln(2m) + R_{path}I_{clique}) \tag{5}$$

By observing equation 5, we can conclude that the complexity function is a quadratic function, which means, there is one and only one extrema. Considering the nature of complexity measure, it's safe to assume that the extrema is a maxima. To find the extrema, we can differentiate the function respect to  $R$  where the function's slope is 0:

$$\frac{dC}{dR} = -2R_{max} + \ln(2m) - I_{clique} + R_{path} = 0 \tag{6}$$

$$R_{max} = \frac{\ln(2m) - I_{clique} + R_{path}}{2} \tag{7}$$

We found  $R_{max}$  where  $C$  reaches its maxima. Substitutes equation 7 into equation 4:

$$C_{max} = (R_{max} - R_{path})(\ln(2m) - R_{max} - I_{clique})$$

$$C_{max} = \left(\frac{\ln(2m) - I_{clique} + R_{path}}{2} - R_{path}\right)(\ln(2m) - \frac{\ln(2m) - I_{clique} + R_{path}}{2} - I_{clique})$$

$$C_{max} = \left(\frac{\ln(2m) - I_{clique} - R_{path}}{2}\right)\left(\frac{\ln(2m) - R_{path} - I_{clique}}{2}\right) \quad (8)$$

$$C_{max} = \frac{(\ln(2m) - I_{clique} - R_{path})^2}{4} \quad (9)$$

Thus, using equation 9, we can define a new measure  $MA_{RI}$ , which is defined by:

$$MA_{RI} = \frac{4(R - R_{path})(I - I_{clique})}{(\ln(2m) - I_{clique} - R_{path})^2} \quad (10)$$

The complexity of  $MA_{RI}$  is equivalent to  $MA_g$ , which can be calculated in  $O(m)$  time.

Some interesting facts about  $MA_{RI}$ :

1. The calculation of  $MA_{RI}$  is solely based on

## 2.3 Potential problems and solutions of different subgraph measures

During the implementation of measures, several problems were found, possible solutions are also given for future discussions.

Different subgraph measures are principally simple, but they are complex to compute, within at least  $O(n^2)$  time[4]. This is not the only problem. An upper-bound of the complexity  $m_{cu} = n^{1.68} - 10$  was introduced by Kim and Wilhelm[4] to normalise the complexity. However, from the simulation, we found that this may not be the actual upper-bound of the different subgraph measures.

The complexity is abnormal for graphs with around 90 edges and 15 nodes as shown in figure 3. This could imply that the upper-bound assumption  $m_{cu}$  is not correct, but there is another possible reason, which is the problem of floating point arithmetic. On most machines today, numbers are represented in binary system[1]. For example, 0.2 is 0.00110011001100110011... in a binary system. This series is infinite, represented by  $1*2^{-3} + 1*2^{-4} + 1*2^{-7} + 1*2^{-8} + \dots$ . For obvious reasons, computer scientists don't want to work with infinite series, therefore, the series is approximated. On a modern computer, the series is usually approximated to 63 digits with 1 digit represents the sign of the number. After approximation, the error could cause the equal operation to fail in programming languages. A well known example is that for modern programming language or machine that operates this numbering system,  $0.2+0.1$  does not equal to  $0.15+0.15$ . As a result, the comparison may cause more number of different subgraph than actual.



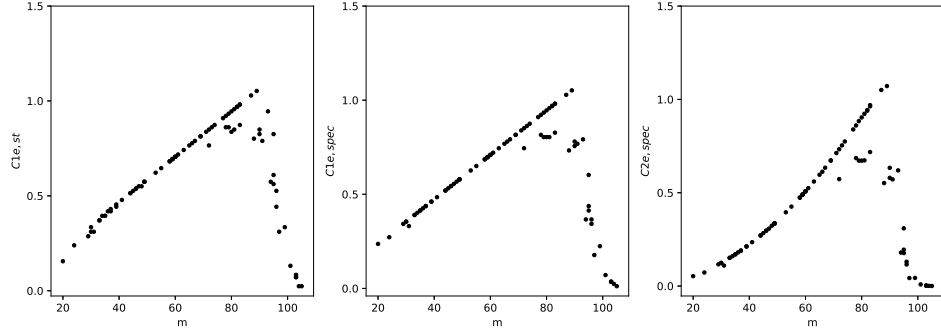


Figure 3: Different subgraph measure of  $G(n, m)$  random graphs, with  $n = 15$ .

The core of different subgraph measure is to compare the cofactor( $C_{1e,st}$ )/spectrum( $C_{1e,spec}$  and  $C_{2e,spec}$ ) of a subgraph. Given the fact that the probability of a decimal number to appear in the spectrums is high and the cofactor will also be very large for a large graph. The comparisons will be inaccurate. There are three possible solutions:

- As suggested, errors will be made when approximated by the machine. An error threshold can be used when comparing spectrums and cofactors. For example, two numbers with relative error less than 1% can also be considered as equal numbers. One disadvantage is the increase of complexity, taking more time and effort to compare the spectrums/number of spanning trees.
- Similarly, numbers can be rounded before comparison to avoid error. This is used in the implementation of different subgraph measures, all cofactors and spectrums are rounded to first 10 significant figures. This solution requires less computation time than first solution. The drawback is that similar graphs can be considered as isomorphic graphs, this also applies to the first provided solution, but with higher accuracy for large graphs. This may still give complexities larger than 1, but it is the best solution considering the effort spent.
- Instead of using  $m_{cu}$  as a normalisation parameter,  $m$  or  $n(n-1)/2$  can be used for one-edge-deleted subgraph complexity and  $\binom{m}{2}$  for two-edges-deleted subgraph complexity. This guarantees the normalisation and avoids the mistake that caused by the first two solutions, but on the other hand, causing the complexity to be different.

A unique problem with  $C_{2e,spec}$  is the value is not properly normalised for small graphs. The upper-bound of  $C_{2e,spec}$  is 0.5 while  $n \leq 7$ . To have an upper-bound at 1, the complexity values have to be scaled by 2. However, scale by 2 will cause the complexity to exceed 1 for larger graphs. Thus, we stuck to the original normalisation and  $C_{2e,spec}$  will have an upperbound at 0.5 for  $n \leq 7$ .

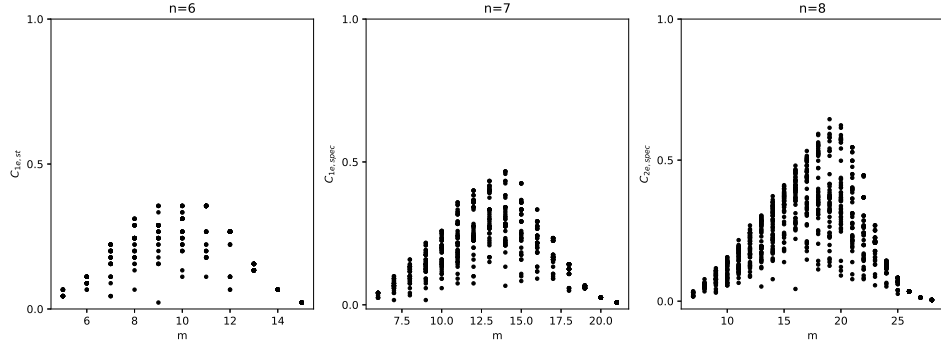


Figure 4:  $C_{2e,spec}$  complexities for  $n = 6, 7, 8$  respectively.

### 3 Result

#### 3.1 Complexity measures on small graphs

As shown in figure 5, we simulated similar results as Kim and Wilhelm in [4]. Except  $C_{2e,spec}$ , as mentioned in section 2.3, there is a scaling problem.

## 4 Conclusion

## References

- [1] 15. *floating point ARITHMETIC: Issues and Limitations*. Date accessed:01/09/2021. URL: <https://docs.python.org/3.8/tutorial/floatingpoint.html>.
- [2] A.L. Barabási. *Network Science*. Cambridge University Press, 2016. ISBN: 9781107076266.
- [3] Anna D. Broido and Aaron Clauset. “Scale-free networks are rare”. In: *Nature Communications* 10.1 (2019). DOI: 10.1038/s41467-019-08746-5.
- [4] Jongkwang Kim and Thomas Wilhelm. “What is a complex graph?” In: *Physica A: Statistical Mechanics and its Applications* 387.11 (2008), pp. 2637–2652. DOI: 10.1016/j.physa.2008.01.015.
- [5] Stanley Milgram. “The small world problem”. In: *Psychology today* 2.1 (1967), pp. 60–67.
- [6] M.E.J. Newman and D.J. Watts. “Renormalization group analysis of the small-world network model”. In: *Physics Letters A* 263.4 (1999), pp. 341–346. ISSN: 0375-9601. DOI: [https://doi.org/10.1016/S0375-9601\(99\)00757-4](https://doi.org/10.1016/S0375-9601(99)00757-4). URL: <https://www.sciencedirect.com/science/article/pii/S0375960199007574>.
- [7] Duncan J. Watts and Steven H. Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393.6684 (1998), pp. 440–442. DOI: 10.1038/30918.

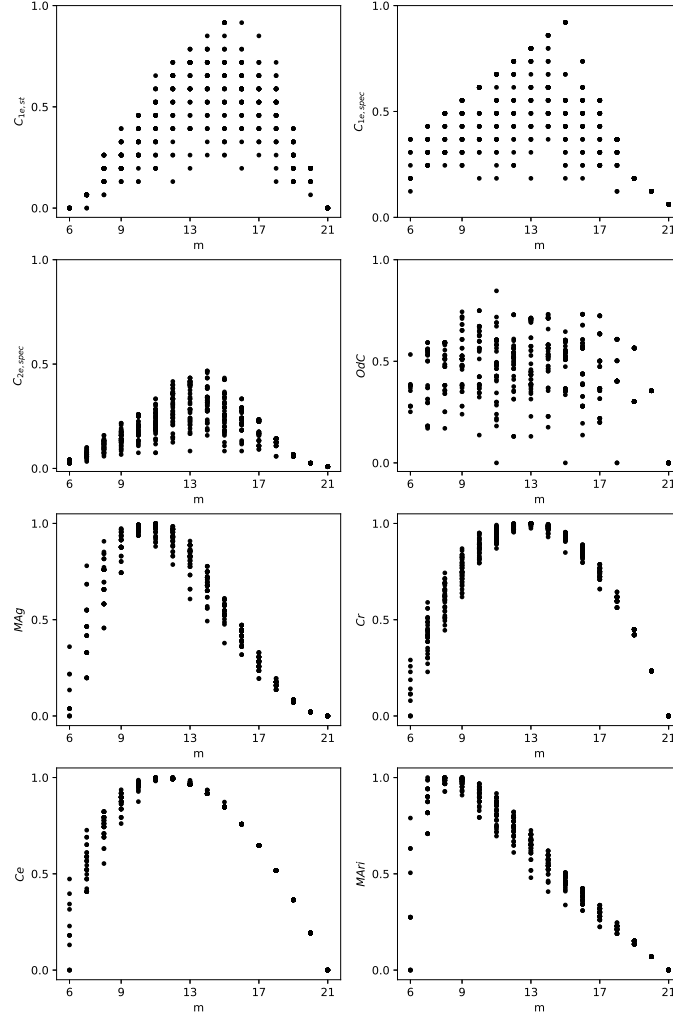


Figure 5: results of implemented methods of graphs with  $n=7$ . Methods from top-left to bottom-right are:  $C_{1e,st}$ ,  $C_{1e,spec}$ ,  $C_{2e,spec}$ ,  $OdC$ ,  $MAg$ ,  $Cr$ ,  $Cr$  and  $MAri$ .