Network Complexity

Thesis Subtitle

Yipei Zhao

A dissertation presented for the degree of
Master of Science

Aston University
BIRMINGHAM UK

MSc Data Analytics
Aston University
United Kingdom
05 September 2021

Contents

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 Small-world

About 50 years ago, a famous study was carried out by Standley Milgram[5] in the interest of this question: how many intermediates are needed to pass a message between two irrelevant or distnaced 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 mordern world. Nevertheless, this number would be smaller than most peoples' 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 probaiblity p the 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.

1.3   Scale-free network

A controversial topic of network science is wether 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 γ is typically in the range $2 < \gamma < 3$. If the degree distrution 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. Initally, 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 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
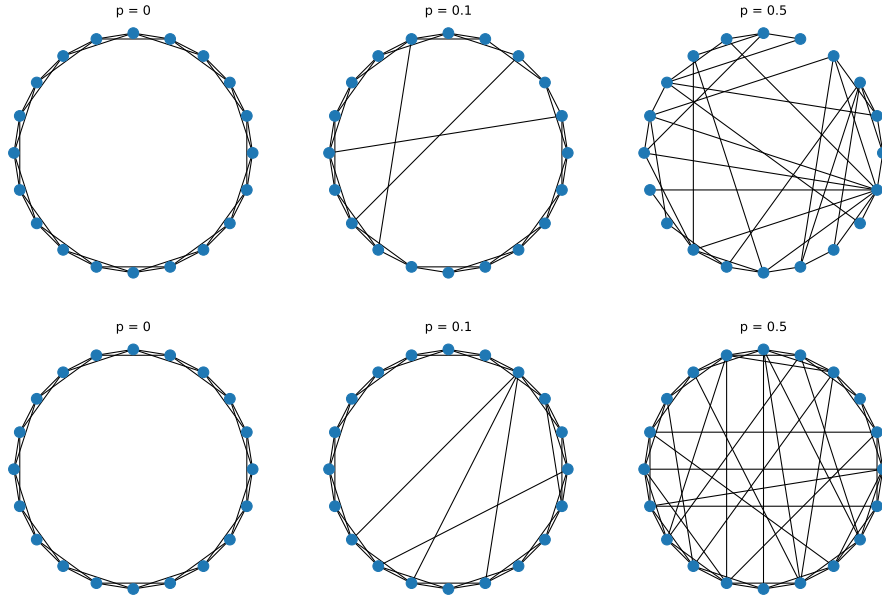
Figure 1: A demonstration of WS model and NW model. The paramters are: $n = 20$, $k = 4$, $p = 0, 0.1, 0.5$.

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 Implemeted methods

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

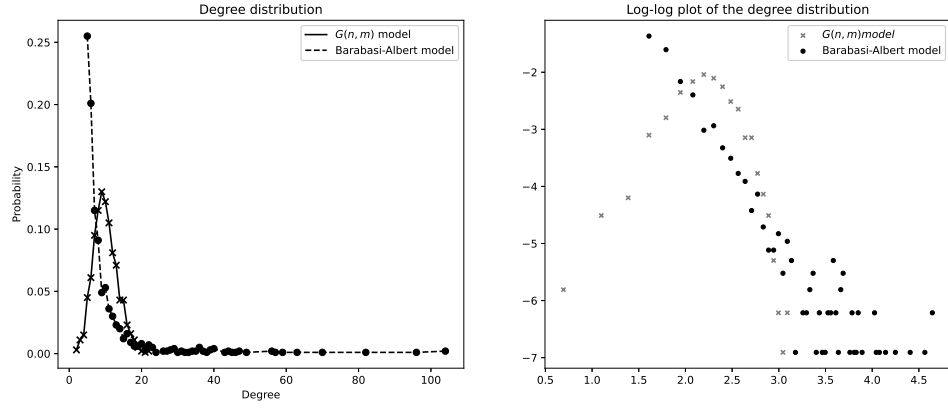- Subgraph measures:

    - $C_{1e,st}$
    - $C_{1e,spec}$

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

- $C_{2e,spec}$

- 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 a normalisation formular,

MAg is defined as[4]:

$$R = \frac{1}{m} \sum_{i,j>i} ln(d_i d_j)$$

$$I = \frac{1}{m} \sum_{i,j>i} ln(\frac{2m}{d_i d_j})$$

$$MA_R = 4(\frac{R - R_{path}}{R_{clique} - R_{path}})(1 - \frac{R - R_{path}}{R_{clique} - R_{path}})$$

$$MA_I = 4(\frac{I - I_{clique}}{I_{path} - I_{clique}})(1 - \frac{I - I_{clique}}{I_{path} - I_{clique}})$$

$$MA_g = MA_R * MA_I$$

(1)

$R_{path}, R_{clique}, I_{path}$ and I_{clique} represent lowest redundacy, 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.

Even though MA_g is well defined and normalised, a problem with the MA_g is it does not assign highest complexities to graphs with the most middle number of edges for small graphs. The highest complexity is reached at $n^{1.5}$, instead of $\frac{n(n-1)}{4}$.
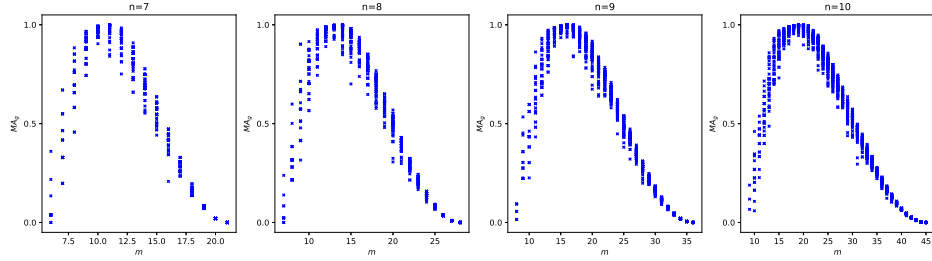

Figure 3: MA_g complexity values of graphs with n=7,8,9,10 nodes

Hence, we may change the normalisation term such that the highest complexity is shifted more towards the middle number of edges. The normalisation term becomes:

$$MA_{RI} = 4(\frac{R - R_{path}}{R_{clique} - R_{path}})(\frac{I - I_{clique}}{I_{path} - I_{clique}})$$

(2)

As showin in figure 4, MA_{RI} and MA_g behave alsmost the same, whereas MA_{RI} assigns higher complexity values to graphs with middle number of edges and averagely higher complexity than MA_g for higher number of edges. The complexity of computation stays the same, which is $O(m)$. MA_{RI} complexity intends to have the nomalization property, which is $0 \leq MA_{RI} \leq 1$, with exceptions.

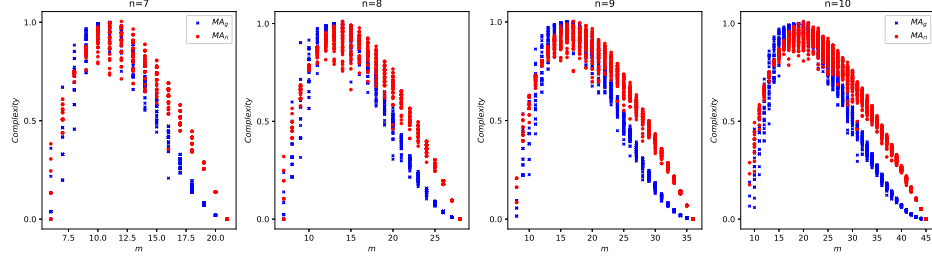During the simulation, a few unnormalization cases were found, they are:

6

Figure 4: Comparison of MA_{RI} and MA_g.

n	m	Highest complexity
6	9	1.0421694413111797
7	11	1.0203660524038531
7	12	1.0042984248515054
8	15	1.0068924792733018

These exceptions are only found for small graphs, for larger graphs, complexities are normalised. Therefore, a solution is proposed.

Observing the equation 2, the extremes are fixed for n, R and I are the variables. Also, R and I are negatively correlated with correlation coefficent equal to -1.

$$R = \frac{1}{m} \sum_{i,j>i} ln(d_i d_j)$$

$$I = \frac{1}{m} \sum_{i,j>i} ln(\frac{2m}{d_i d_j})$$

$$I = \frac{1}{m} \sum_{i,j>i} ln(2m) - \frac{1}{m} \sum_{i,j>i} ln(d_i d_j)$$

$$I = ln(2m) - R$$

(3)

Thus, if the maximum of $\sum_{i,j>i} ln(d_i d_j)$ can be found, the maxmimum of $MAri$ can be found as well. The maximum can be found in two ways:

- Creates all graphs with the respective n, and find the maximum complexity, and divide all valeus by the maximum complexity.

- As R and I are negatively correlated with coefficent -1, we can find the maximum of $MAri$. Given the fact $R_{min} = R_{path} = ln(2m) - I_{max} = ln(2m) - I_{path}$ and $R_{max} = R_{clique} = ln(2m) - I_{min} = ln(2m) - I_{clique}$. By varying the value of R between R_{min} and R_{max}, MA_{RImax} can be found.

We only suggest to use the solutions for small graphs, not only the problem only occurs for small graphs, but also the complexity will be largely increased for large
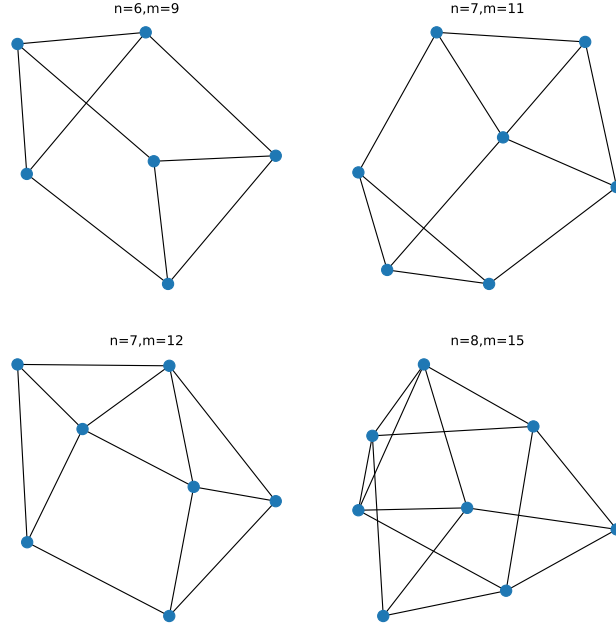
7

Figure 5: Graphs with highest MA_{RI} complexity.

graphs.

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 6. 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, repre-
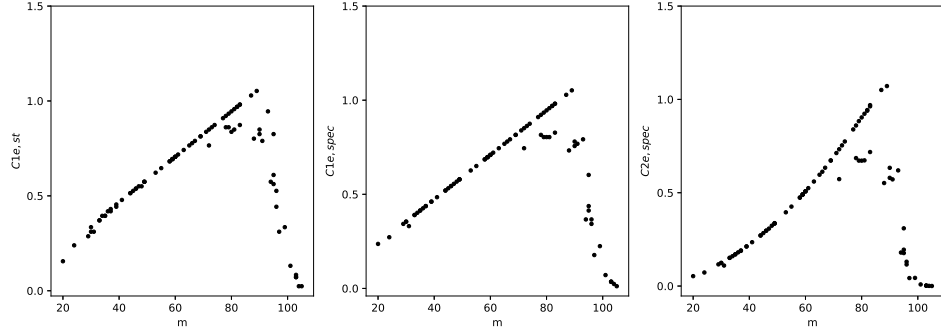
Figure 6: Different subgraph measure of $G(n, m)$ random graphs, with $n = 15$.

sented by $1*2^{-3}+1*2^{-4}+1*2^{-7}+1*2-8....$ For obvious reasons, computer scientists don't want to work with infinite series, therefore, the series is approximated. On a modern compueter, 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.

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 proabbility of a decimal number to appear in the sprectrums 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 approxiamted 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 gives complexities larger than 1, but it is the best solution considering the efforst 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 gurantees the normalisation and avoid the mistake

9

that caused by the first two solutions, but on the other hand, causing the complexity to be different.

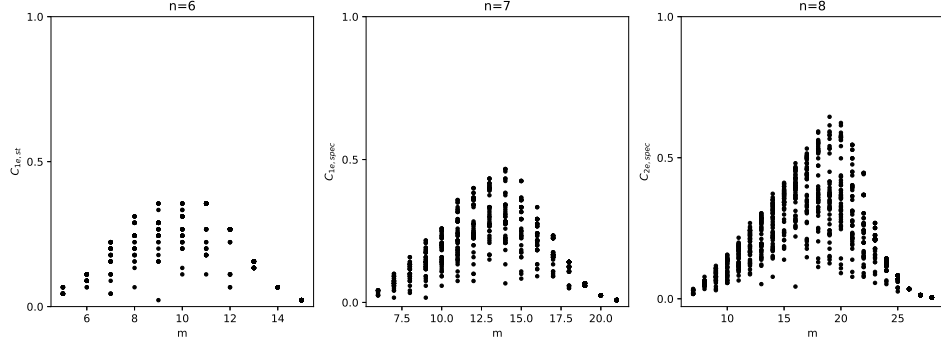A unique problem with $C_{2e,spec}$ is the value is not properly normalised for small graphs.

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

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 sticked to the original normalisation and $C_{2e,spec}$ will have an upperbound at 0.5 for $n \leq 7$.

3 Result

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. 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.
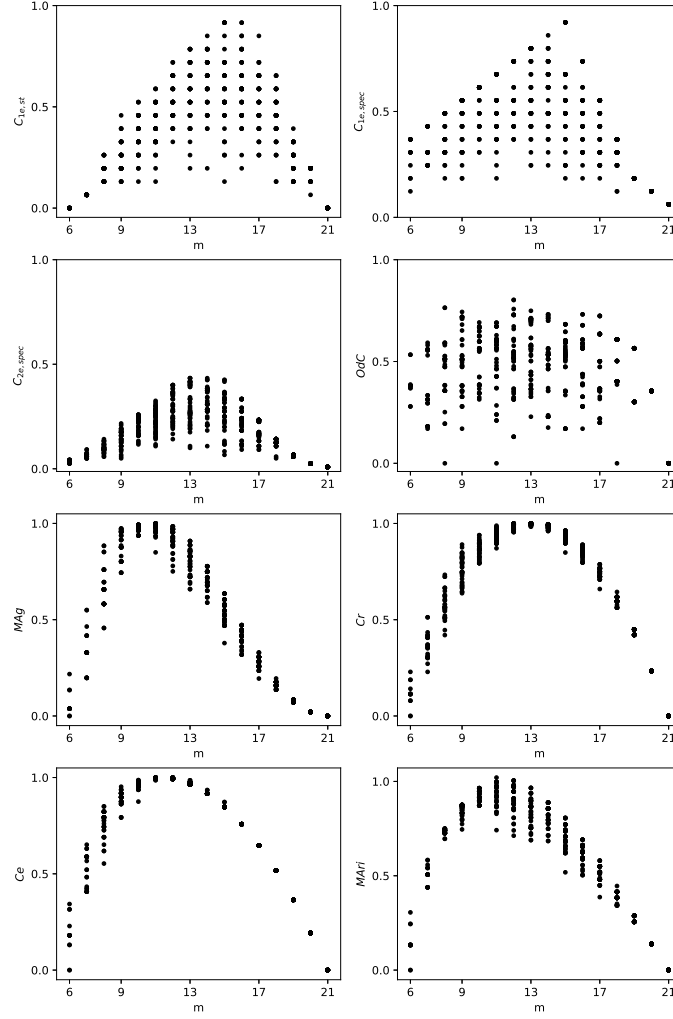
Figure 8: results of implemented methods of graphs with n =7. Methods from top-left to bottom-right are: $C_{1e,st}$, $C_{1e,spec}$, $C_{2e,spec}$, OdC, MAg, Cr, Cr and $MAri$.