Exploring methods to determine similarity between simulated and real-life social networks

Martin Measic[1]

[1] Faculty of Information Technology, Monash University, Australia,
mmea0003@student.monash.edu

Abstract. Graphs are everywhere. Determining similarity between two graphs may be as simple as inspecting the two graphs and observing their structures when they are small enough. However, when the number of vertices in a graph becomes extremely large, relying solely on visual inspection is neither practical nor reliable. This study focuses on the methods used to determine similarity between simulated and real-life social graphs. A *Baseline* model, which was formed by trial and error, and a *Random* model were both compared against a real-life social network on various graph metrics. The results found that both models were extremely similar to the real-life social network across all metrics tested. It was established that various metrics, which do not rely on visual inspection, could be used to test graphs for similarity.

Keywords: Graphs, social networks, network similarity, agent-based modeling, random graphs

1 Introduction

Graph-based networks are used to model various real-life systems (Leishman & Green, 2008). From modeling public transportation and virus spread to simulating human interaction, graphs provide a natural representation of the complex structures and patterns observable around the world. It is therefore important to ensure that graphs accurately depict the systems they are trying to represent – i.e., that the graphs are similar.

The main objective of this study is to determine whether a computer simulation can model social networks. In particular, the aims are to create a computer model which is representative of a real-life social network and to understand which metrics can be used to test graphs for similarity. Various graph metrics such as degree distribution, clustering coefficient, and diameter have been used in other studies to test for graph similarity (Kim & Leskovec, 2011; Madey et al., 2003; Singer et al., 2009). This study will focus on expanding the metrics which have already been used to test graphs for similarity, by considering a greater range of metrics.

Understanding which aspects make a social network similar to its graph representation will provide an important foundation for similarity testing. In turn, this will enable the creation of more accurate models (Stocker et al., 2014).

2 Background

2.1 Modeling social networks as graphs

There has been much research on how to model social networks as graphs. In its simplest form, a graph can represent a social network by modeling individuals as vertices and relationships as pairwise edges (Green et al., 2007). Most existing models of social networks focus on modeling structures while ignoring the features and properties of the vertices (Kim & Leskovec, 2012). Vertices alone cannot capture the complexity of the individual – their likes, their dislikes, their humanity. Just as real social networks rely on individuals interacting based on their human qualities, a simulated network must also take these attributes into account. One solution to this problem is to represent vertices as agents – each of which is capable of having numerous interests with binary or many states (Kim & Leskovec, 2011).

While modeling vertices as agents provides depth to individuals, it does not provide any reasoning for how relationships should be maintained. Social psychology suggests that humans tend to associate with those who share similar beliefs, interests, and values at a much higher rate than among dissimilar people – a principle known as homophily (Fu et al., 2012; McPherson et al., 2001). Likewise, this belief also suggests that dissimilar individuals will break ties faster than those that are similar. In social networks, vertices that have attributes are expected to be active agents that have a direct impact on those around them (Borgatti et al., 2018). A study by Madey et al. in 2003, which used an agent-based model for simulating friendship in social networks, observed self-organized community structures emerging in the models produced. In the study, friendship formation was based on the frequency of encounters and mutual interest. We now introduce a process known as *Dual Phase Evolution*, which has been extensively documented in other simulations of social networks (Green, 2014).

2.2 *Dual Phase Evolution* and its role in emergent structures

Dual Phase Evolution is a self-occurring process that leads to the organization of individuals in an existing graph network (Green, 2014; Leishman et al., 2009; Stocker et al., 2014). In *Dual Phase Evolution*, interaction fluctuates between two different phases: a *local* phase, where existing relationships are maintained and individuals interact within their local circle, and a *global* phase, where new relations are formed outside of existing communities. Consider what happens when the number of interests or the number of states an interest has is changed. *Dual Phase Evolution* has shown different network structures form depending on whether single-interest or multi-interest agents are used. The same study by Madey et al. in 2003 mentioned previously found that macroscopic interactions between individual agents lead to global properties emerging. The idea that a small local change can lead to changes in the structural and topological characteristics of a network is an idea supported by *Dual Phase Evolution*. When agents have single interests, *Dual Phase Evolution* leads to tightly knit clusters as from individual to individual there is only one possibility: either they share the same interest or not. More fascinating, though, is what happens when agents have multiple interests. The ability for individuals to share many things in common results in the kind of structures and patterns observable in real-life social networks: long tree-like chains and branches which contain relatively few loops.

2.3 Determining similarity

Despite there being much research in the area of graph similarity, the definition of two graphs being 'similar' remains quite broad. Are two graphs similar if they look the same? If they have the same observable structures? Previous studies which have focused on analyzing the similarities between real-life social networks and computer models have shown similarities to exist based on visual inspection (Bearman et al., 2004; Green 2014). However, since graph-based networks can have infinitely many different representations, determining similarity based on visual inspection alone is neither feasible nor reliable.

One way to determine similarity is to test for graph isomorphism. Two graphs are isomorphic if there is a bijective mapping that preserves the relationships between vertices (Champin & Solnon, 2003). However, to consider true graph isomorphism, the number of vertices in a graph must be identical. This poses an issue when comparing graphs that do not have identical vertex counts. While we can test subgraphs for isomorphism, the algorithms to do so are exponential in time complexity (Koutra et al., 2011). It is therefore infeasible to conduct hundreds, let alone tens of thousands of repeat trials.

Edit distance, which is the minimum number of operations needed to transform one graph to another, is another way to measure similarity (Li et al., 2019). However, like graph isomorphism, it is exponential in time complexity (Zeng et al., 2009).

Other algorithms for determining similarity have been established which are more efficient. For instance, Koutra et al. (2011) detail an algorithm where a degree of similarity between two graphs is measured as a real number from 0 to 1. This idea of assigning a similarity index has also been used in other studies, where a similarity score for the vertices of two graphs was calculated by structural similarity of local neighborhoods (Zager & Verghese, 2008).

While these algorithms and methods are sophisticated and powerful in determining similarity, a function that assigns to two graphs a single-valued similarity index is not informative of the underlying processes occurring in the graphs. Determining similarity in this study will therefore focus on analyzing the metrics used in these studies more closely, to obtain a finer level of detail.

2.4 Bearman's case study

In 2004, Bearman, Moody, and Stovel conducted a study of the social and romantic network found at Jefferson High School. Of the 832 respondents interviewed, 573 reported being involved in some form of relationship with another student in the past 18 months.

A previous study by Green in 2014 found that a computer-simulated social network with multi-interest agents operating under the process of *Dual Phase Evolution* was visually similar to the graph of the social network present at Jefferson

High School: both had a tree-like structure with long chains and relatively few smaller loops. This observed similarity will be further explored by generating social networks and comparing them to this real-life model.

3 Methodology

The general method used in the study was to create the simulation, collate various metrics during the running of the simulation, and then compare these metrics against the social network studied in Bearman's study.

3.1 Creating the simulation

The simulation was written in version 17 of the Java SE Platform language using an agent-based model represented by a graph. This approach was chosen due to its simplicity and power in network representation (Green et al., 2007). Each *simulation* consisted of a *graph* with *n* vertices and *n* agents. Each agent had a randomized list of *m* interests with an attribute value assigned to each interest. An attribute value was simply taken to be a random integer in some range. The simulation was run for a specified number of steps, following the process of *Dual Phase Evolution*. The *global* phase was conducted first, followed by the *local* phase. During the *global* phase, agents would create random links with others based on some set probability. During the *local* phase, agents would break links with others if the number of interests they shared did not meet some specified threshold. Two agents had interest *i* in common if both agents had identical attribute values at that interest. Table 1 below explains the parameters used.

agentNumber: the number of agents used in the simulation
numberOfInterests: the number of interests an agent had
statesPerInterest: the number of states a single interest could take
levelOfInterestNeeded: the number of interests agents needed in common to maintain links
maxEdgesPerNode: the maximum number of links a single agent could have
maxGlobalEdges: the maximum number of links in the whole graph
stepsToRun: the number of times the local and global phases were repeated
probRandomEdges: the probability of creating random edges

Table 1. An explanation of the parameters used in the simulation.

3.2 Models used

Two separate models were considered to compare against the graph of the social network at Jefferson High School (which from now will be referred to in short as the *Bearman* model). The first is the *Random* model, which used the Erdős-Rényi model for generating random networks (see Algorithm 1 below). The second is the *Baseline* model, which used parameters that produced graphs that looked visually similar to

Bearman. The parameters for the *Baseline* model were chosen by experimentation (see Table 2 below). Both models had the number of vertices set as the number of participants in Bearman's case study (832) and the probability of creating edges set at the density of *Bearman*. The two models were both simulated for 10,000 repeat trials, and various metrics were recorded. Isolated nodes were removed from the graphs to mimic Bearman's treatment of the data.

```
probability = (2.0*477)/(832*831)
for i in range [0, n):
    for j in range [i + 1, n):
        randomValue = random decimal in range [0, 1]
            if randomValue < probability:
                add an undirected edge between i and j
```

Algorithm 1. The Erdős-Rényi model for generating random networks (Porekar, 2002). The probability for creating edges was set at the density of *Bearman.* It is important to note that the density of *Bearman* used here considers a vertex count of 832 instead of 573. This assumption ensures an accurate density is used, as Bearman removed isolated nodes from his study.

agentNumber: 832
numberOfInterests: 10
statesPerInterest: 2
levelOfInterestNeeded: 7
maxEdgesPerNode: ∞
maxGlobalEdges: ∞
stepsToRun: 6
probRandomEdges: (2.0*477)/(832*831)

Table 2. The parameters used in *Baseline*. These were chosen through experimentation until the graphs produced looked similar to *Bearman*. probRandomEdges was taken to be the edge density of *Bearman*. Multi-interest agents with binary attributes (YES/NO) were used.

3.3 Recording metrics

Various metrics were recorded during the running of each trial to create a distribution of the results obtained. Each model was simulated 10,000 times to gain an accurate depiction of the mean and standard deviation of each metric calculated. These metrics were then compared against *Bearman*. The full list of metrics and how they were calculated can be found below. Formulas 1 through 7 detail the method used to calculate assortativity coefficient, average degree, average clustering coefficient, the number of cycle basis components, density, diameter, and remaining vertex count, respectively. Algorithm 1 details the algorithm used to calculate the number of connected components.

$$\rho(X,Y) = \frac{E[(X - E[X])(Y - E[X])]}{\sqrt{E[(X - E[X])^2]E[(Y - E[Y])^2]}}$$

Formula 1. Assortativity coefficient. This definition is based on the correlation between random variables – X and Y are the excess degree at the end of a link (Noldus & Van Mieghem, 2015).

$$AverageDegree\big(G(V,E)\big) = \frac{2|E|}{|V|}$$

Formula 2. Average degree. $G(V,E)$ is a graph with vertices V and edges E (Lizardo & Jilbert, 2020a).

$$C_i = \frac{2L_i}{k_i(k_i - 1)}$$

$$[C] = \frac{1}{N}\sum_{i=1}^{N} C_i$$

Formula 3. Average clustering coefficient. The clustering coefficient of a node i is denoted by C_i. k_i refers to the degree of node i and L_i is the number of edges between the k_i neighbors of node i. $[C]$ is the average clustering coefficient of the entire graph (Pasta, 2015).

$$CycleBasis\big(G(V,E)\big) = |E| - |V| + c$$

Formula 4. Number of cycle basis components. $G(V,E)$ is a graph with vertices V and edges E. c represents the number of connected components in G (Diestel, 2016).

$$Density\big(G(V,E)\big) = \frac{2|E|}{|V|^2 - |V|}$$

Formula 5. Density. $G(V,E)$ is a graph with vertices V and edges E (Lizardo & Jilbert, 2020b).

$$\delta = \max\{s(i,j)\}$$

Formula 6. Diameter. $s(i,j)$ denotes the number of edges in the shortest path from vertex i to j. The max function calculates the longest path among all shortest paths between every pair of vertices i and j in the graph (Grossmann & Flitter, 2016). The shortest path between two points was calculated using the Breadth-First search traversal algorithm (Gupta, 2020). ‘Diameter’ was taken to be the maximum diameter among all connected components.

$$RemainingVertexCount\big(G(V,E)\big) = |V| - |I|$$

Formula 7. Remaining vertex count. $G(V,E)$ is a graph with vertices V and edges E. I denotes the isolated vertices in G ($I \subseteq V$). The number of isolated vertices in G was calculated by counting the number of vertices with degree 0.

```
function numberOfConnectedComponents(G(V, E)):
    visited = Boolean array of size |V| initialized
        to false values
    count = 0
    for i in range [0, |V|):
        if visited[i] is false:
            dfs(i, visited)    //dfs sets all vertices reachable
                from i as true in the visited array
            count = count + 1
    return count
```

Algorithm 2. The algorithm used to calculate the number of connected components (Srivastava, 2021). *dfs* is a call to a function which performs a standard Depth-First search (Garg, 2021).

4 Results and discussion

The results of the simulations conducted show that on all metrics tested, *Bearman* was well within one standard deviation of the mean in both *Random* and *Baseline*. The standard deviations were quite small for the majority of the metrics recorded which indicates low variation. It was quite astonishing to see the simulated networks produced were overwhelmingly similar to *Bearman*. Perhaps the most surprising finding of the study was that the *Random* model was not only quite similar to *Baseline*, but also quite similar to *Bearman* as well. A plausible explanation for the *Random* model being so similar to *Bearman* might be due to the fact the probability of creating edges was set to such a low value. As a result, there was a lack of denseness which one would usually expect to see in a random network. Sample visualizations of the graphs produced, including the graph from Bearman's original study, are provided in Appendix A.

4.1 Results and discussion of the main metrics

The following graphs present the results of metrics tested. GraphPad Prism version 9.2.0 for macOS was used to calculate the statistics and draw the graphs. The horizontal axis plots the values of each metric, and the vertical axis plots the results of the two models. The circles represent the mean value present in the 10,000 trials conducted and the endpoints of the solid lines indicate values that lied within one standard deviation of the mean. The dashed line represents the value Bearman's model had for that metric. Do note that the domain of the horizontal axis is quite small in many graphs as there was little variation in many of the results obtained.

The results for assortativity coefficient (Figure 1) show that *Random*, *Baseline*, and *Bearman* all displayed characteristics of non-assortativity. There was little variation in the results obtained as all were very close to 0. These results are expected for *Baseline* and *Bearman* as nodes should cluster based on interests rather than degree.

In terms of average degree (Figure 2), both the *Baseline* and *Random* models had mean values which were extremely close to the average degree of *Bearman*, with a difference of less than 0.05. *Random's* mean was slightly closer to *Bearman* – however, by an amount so small it can be considered negligible.

There was almost no clustering present in *Random*, *Baseline*, or *Bearman* (Figure 3). The average for *Baseline* was extremely close to *Bearman*, with a recorded difference of 0.00008. However, we can also consider the average of *Random* to be within range of *Bearman*, since the difference was also quite small (0.0007072).

Figure 4 reveals there were very few cycles present in most graphs produced. Both the *Baseline* and *Random* models were well within the number of cycles in *Bearman*. *Baseline's* average was slightly higher and closer to *Bearman* than *Random* and had a larger range.

The results for density display very little variation (Figure 5). Both *Baseline* and *Random* had almost identical spreads of density – which would be expected considering both models were given the same probability of creating edges. Note that the density of all graphs was calculated after isolated nodes were removed to ensure consistency with Bearman's treatment of data.

The diameter of both models was well within range of *Bearman* (Figure 6). The spread of *Baseline* and *Random* were again quite similar, however with *Baseline's* average being slightly closer to *Bearman*.

Figure 7 graphs the remaining vertex count. Since each trial in both models began with 832 vertices (the same number of participants in *Bearman*), it was quite incredible that the average number of non-isolated vertices remaining was so close to *Bearman* (573). *Baseline's* average was slightly closer to *Bearman*.

In terms of the number of connected components, both models had mean values similar to *Bearman* (Figure 8). The distribution of *Baseline* was slighter lower, indicating fewer pairs and triads than in *Bearman*. *Random* was closer to *Bearman* on this metric but not by a large amount.

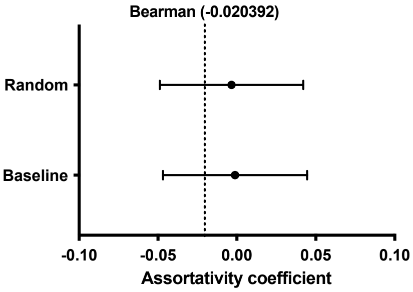

Figure 1. The assortativity coefficient of a graph measures the relationship between pairs of linked nodes. The values range from -1 to 1, with positive values indicating that nodes of similar degree tend to link with other nodes of similar degree, and negative values indicating that nodes of

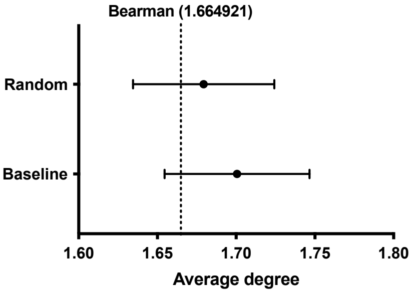

Figure 2. The average degree of a graph measures the average number of edges per node in a graph (Lizardo & Jilbert, 2020a).

dissimilar degree tend to link together. A value of 0 indicates non-assortativity (Noldus & Van Mieghem, 2015).

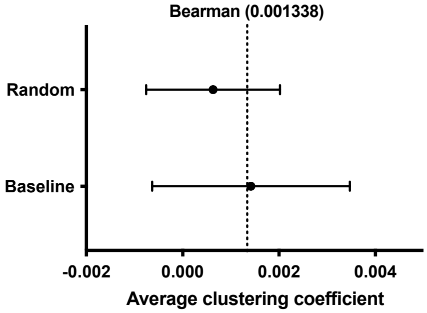

Figure 3. The clustering coefficient of a node is a local measure taken to be the degree to which nodes tend to cluster together. The average clustering coefficient is the average clustering coefficient of all nodes in the graph. The values range from 0 to 1, with 0 indicating few connections in the neighborhood and 1 indicating the neighborhood is fully connected (Pasta, 2015). Note that since the standard deviation is larger than the mean in both models, the left endpoint appears below 0.

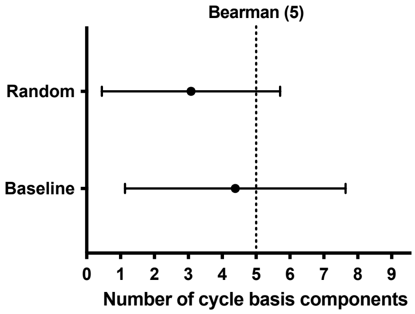

Figure 4. The number of cycle basis components in a network is a measure of the difference in edge count between the total edge count of each connected component and the spanning forest of the graph (Diestel, 2016). In essence, it provides a count of the cycles present in the graph.

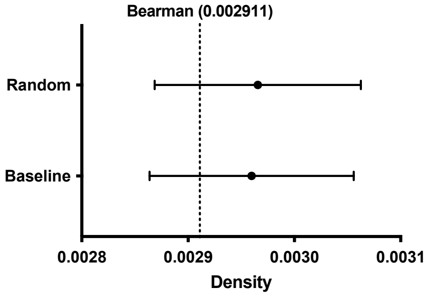

Figure 5. The density of a graph is a measure of how many edges exist between nodes compared to the total number of possible edges between nodes (Lizardo & Jilbert, 2020b). A dense graph will have many edges whereas a sparse graph will have very few edges.

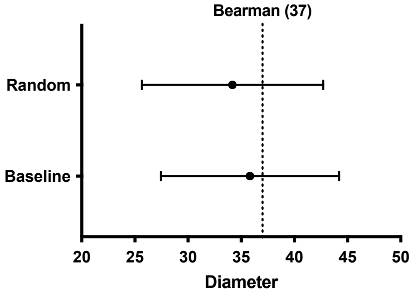

Figure 6. The diameter of a graph is a measure of the longest shortest path present in the graph (Grossmann & Flitter, 2016). That is, among all the shortest paths between each pair of vertices in the graph, the diameter is taken to be the longest path.

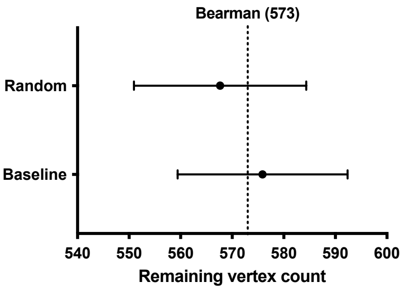

Figure 7. The remaining vertex count was taken to be the number of vertices left in the graph after isolated vertices were removed.

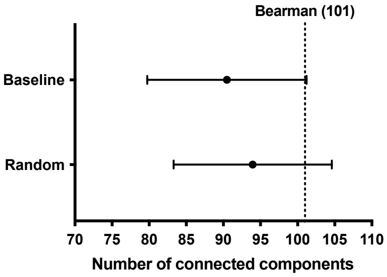

Figure 8. The number of connected components in a graph is a count of the number of subgraphs in which each node in the component is reachable to other nodes in the same component (Srivastava, 2021). In essence, it is a measure of the number of disjoint pieces a graph has.

4.2 Network properties and topologies

The graphs were also tested for properties of small-world and scale-free networks.

The defining properties of a small-world network is that it has a high clustering coefficient, a small diameter, has many high degree nodes in its degree distribution, and has most nodes connected to each other via short paths (Porekar, 2002; Porter, 2012). The *Baseline* and *Random* models did not exhibit such characteristics, and neither did *Bearman*. Both models displayed an extremely low average clustering coefficient, a large average diameter, and a very low average degree. The degree distributions of both models also showed that the vast majority of nodes had a degree of one or two only, with an extremely low number of nodes having a degree greater than five among all trials conducted.

The defining properties of a scale-free network is that its degree distribution follows a power law (Hidalgo & Barabasi, 2008). The three graphs below show the degree distributions of *Bearman* (Figure 9), and a single sample graph taken from *Baseline* (Figure 10) and *Random* (Figure 11). The two samples were each randomly chosen from the 10,000 graphs produced for *Baseline* and *Random*, respectively. While it does appear that all three models fit the lines provided, as stated in Bearman's original study: "it is unlikely that the observed degree distribution is in fact scale free" (Bearman et al., 2004). This is because the *Baseline* model (and Bearman's original study) showed that students did not base their friendships on the number of friends they already had, but rather, on how many interests they had in common with others. Scale-free networks contain many high-degree nodes, which have degrees that greatly exceed the average degree (Meghanathan, 2015). This characteristic was also not observed in the graphs produced nor in Bearman's original study. While it would usually be expected for a random graph's degree distribution to follow a binomial or

Poisson distribution (Porekar, 2002), the *Random* model here follows the same distribution as *Bearman* and *Baseline*. An explanation for this is most likely because the number of nodes in the graph was not large and the probability of creating edges was low. Even though the graphs are unlikely scale-free, we can interpret the similar patterns observed in their degree distributions as another indication of similarity between the models.

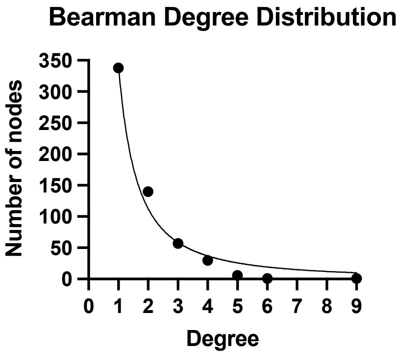

Figure 9. The degree distribution of *Bearman*. The slope of the curve is -1.606 and its R^2 value is 0.9821.

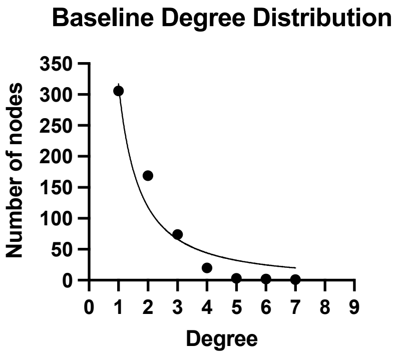

Figure 10. The degree distribution of a sample network taken from the *Baseline* model. The slope of the curve is -1.424 and its R^2 value is 0.9374.

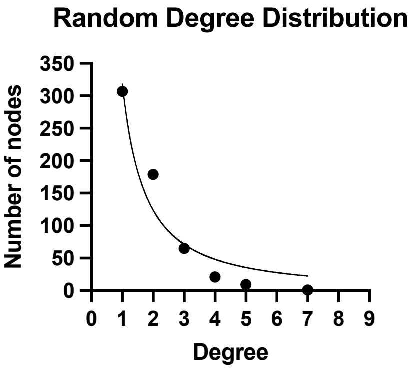

Figure 11. The degree distribution of a sample network taken from the *Random* model. The slope of the curve is -1.364 and its R^2 value is 0.9314.

4.3 Limitations and future work

In Bearman's original study, the students interviewed were only asked to identify at most "three individuals with whom they had a nonromantic sexual relationship in the past 18 months" (Bearman et al., 2004). By setting this limitation at only three outgoing edges, even though in the end all edges were considered undirected, the graph produced may not have captured the total number of relationships an individual could have. Future studies could recreate this real-life study with no such limit, to determine whether the results obtained by simulation are reproducible at larger scales.

It is also important to realize that the real world is much more complex than any model that simulates it. It was out of the scope of this study to simulate the intricacies and depth of human interaction and the way in which relationships form. Nonetheless, this project has provided great importance in understanding both social relationships and the methods used to test network similarity.

5 Conclusion

The focus of this research has been to determine whether a computer model can recreate the patterns and structures observable in real-life social networks. While it was expected that the creation of some model, which we called the *Baseline* model, would be similar to a real-life model, it was not expected for the *Random* model to also be similar to a real-life model. The major, unexpected finding of this study is how strikingly similar the *Random* model was to *Bearman* on most metrics. As detailed in the discussion section, this similarity is most likely due to the fact the probability of creating edges was set at a very low value. Nonetheless, the conclusion is that it is possible to recreate real-life social models, in particular the social model observed in Jefferson High School by Bearman and colleagues, with high accuracy through the process of simulation. Various metrics have also been identified that provide a solid framework and means for similarity testing. These metrics provide strong evidence for testing graph similarity that extend far beyond simple visual inspection and can be used in many fields which rely on similarity testing. In turn, this will enable the creation of more accurate networks which are more representative of the real-life systems they are supposed to model.

6 References

Bearman, P. S., Moody, J., & Stovel, K. (2004). Chains of affection: The structure of adolescent romantic and sexual networks. *American Journal of Sociology*, *110*(1), 44–91. https://doi.org/10.1086/386272

Borgatti, S. P., Everett, M. G., & Johnson, J. C. (2018). *Analyzing Social Networks*. Sage.

Champin, P.-A., & Solnon, C. (2003). Measuring the Similarity of Labeled Graphs. *Case-Based Reasoning Research and Development*, 80–95. https://doi.org/10.1007/3-540-45006-8_9

Diestel, R. (2016). *Graph Theory* (5th ed.). Springer.

Fu, F., Nowak, M. A., Christakis, N. A., & Fowler, J. H. (2012). The Evolution of Homophily. *Scientific Reports*, *2*(1). https://doi.org/10.1038/srep00845

Garg, P. (2021). *Depth First Search*. HackerEarth. Retrieved from https://www.hackerearth.com/practice/algorithms/graphs/depth-first-search/tutorial/.

Green, D. G. (2014). *Of Ants and Men: The Unexpected Side Effects of Complexity in Society*. Springer.

Green, D. G., Leishman, T. G., & Sadedin, S. (2007). The emergence of social consensus in Boolean Networks. *2007 IEEE Symposium on Artificial Life*. https://doi.org/10.1109/alife.2007.367823

Grossmann, T., & Flitter, H. (2016). Diameter of a graph. Retrieved from http://www.gitta.info/Accessibiliti/en/html/StructPropNetw_learningObject2.html.

Gupta, A. (2020). *Shortest path in an unweighted graph*. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/shortest-path-unweighted-graph/.

Hidalgo, C. A., & Barabasi, A.-L. (2008). Scale-Free Networks. *Scholarpedia*, *3*(1), 1716. https://doi.org/10.4249/scholarpedia.1716

Kim, M., & Leskovec, J. (2011). *Modeling Social Networks with Node Attributes using the Multiplicative Attribute Graph Model*. arXiv.org. Retrieved from https://arxiv.org/abs/1106.5053.

Kim, M., & Leskovec, J. (2012). Multiplicative Attribute Graph Model of Real-World Networks. *Internet Mathematics*, *8*(1-2), 113–160. https://doi.org/10.1080/15427951.2012.625257

Koutra, D., Parikh, A., Ramdas, A., & Xiang, J. (2011). (rep.). *Algorithms for Graph Similarity and Subgraph Matching*. Retrieved from http://www.stat.cmu.edu/~aramdas/reports/DBreport.pdf.

Leishman, T. G., Green, D. G., & Driver, S. (2009). Self-organization in Simulated Social Networks. *Lecture Notes in Computer Science*, 150–156. https://doi.org/10.1007/978-3-642-02276-0_15

Leishman, T., & Green, D. G. (2008). Formation of network modules by dual phase evolution. In *2008 World Automation Congress*. Waikoloa, HI; IEEE.

Li, Y., Gu, C., Dullien, T., Vinyals, O., & Kohli, P. (2019). Graph Matching Networks for Learning the Similarity of Graph Structured Objects. In *Proceedings of the 36th International Conference on Machine Learning*.

Long Beach, California. Retrieved from http://proceedings.mlr.press/v97/li19d/li19d.pdf.

Lizardo, O., & Jilbert, I. (2020a). *2.7 Average Degree: Social Networks: An Introduction.* Social Networks: An Introduction. Retrieved from https://bookdown.org/omarlizardo/_main/2-7-average-degree.html.

Lizardo, O., & Jilbert, I. (2020b). *2.9 Density: Social Networks: An Introduction*. Social Networks: An Introduction. Retrieved from https://bookdown.org/omarlizardo/_main/2-9-density.html.

Madey, G., Gao, Y., Freeh, V., Tynan, R., & Hoffman, C. (2003). Agent-Based Modeling and Simulation of Collaborative Social Networks. In *AMCIS 2003 Proceedings*. Retrieved from http://aisel.aisnet.org/amcis2003/237.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, *27*(1), 415–444. https://doi.org/10.1146/annurev.soc.27.1.415

Meghanathan, N. (2015). *Scale-Free Networks*. Lecture. Retrieved from http://www.jsums.edu/nmeghanathan/files/2015/08/CSC-641-Fall2015-Module-5-Scale-free-Networks.pdf.

Noldus, R., & Van Mieghem, P. (2015). Assortativity in Complex Networks. *Journal of Complex Networks*, *3*(4), 507–542. https://doi.org/10.1093/comnet/cnv005

Pasta, M. Q. (2015). Calculating Clustering Coefficient. Retrieved from http://qasimpasta.info/data/uploads/sina-2015/calculating-clustering-coefficient.pdf.

Porekar, J. (2002). *Random Networks*. Department of Theoretical Physics. Retrieved from http://www-f1.ijs.si/~rudi/sola/Random_Networks.pdf.

Porter, M. (2012). Small-World Network. *Scholarpedia*, *7*(2), 1739. https://doi.org/10.4249/scholarpedia.1739

Singer, H. M., Singer, I., & Herrmann, H. J. (2009). Agent-based model for Friendship in Social Networks. *Physical Review E*, *80*(2). https://doi.org/10.1103/physreve.80.026113

Srivastava, S. (2021). *Program to count number of connected components in an undirected graph*. GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/program-to-count-number-of-connected-components-in-an-undirected-graph/.

Stocker, R., Green, D. G., & Sadedin, S. (2014). Complexity and Human Society. In *Networks in Society: Links and Language* (pp. 81–105). essay, Pan Stanford Publ.

Zager, L. A., & Verghese, G. C. (2008). Graph similarity scoring and matching. *Applied Mathematics Letters*, *21*(1), 86–94. https://doi.org/10.1016/j.aml.2007.01.006

Zeng, Z., Tung, A. K., Wang, J., Feng, J., & Zhou, L. (2009). Comparing Stars: On Approximating Graph Edit Distance. *Proceedings of the VLDB Endowment*, *2*(1), 25–36. https://doi.org/10.14778/1687627.1687631

Appendix A

This appendix consists of a visualization of the graph produced in Bearman's case study (Figure A1) and three sample graphs from both the *Baseline* (Figure A2, Figure A3, Figure A4) and *Random* (Figure A5, Figure A6, Figure A7) models. They were produced using Pajek – a program for large network analysis.

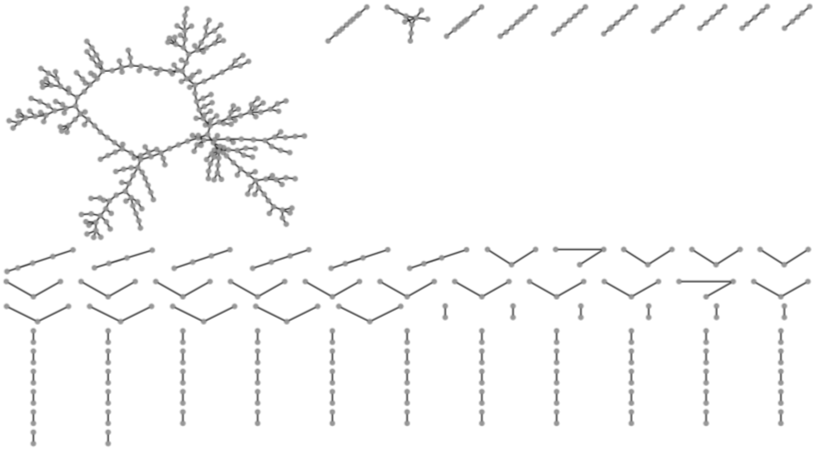

Figure A1. A visualization of the graph produced in Bearman's case study.

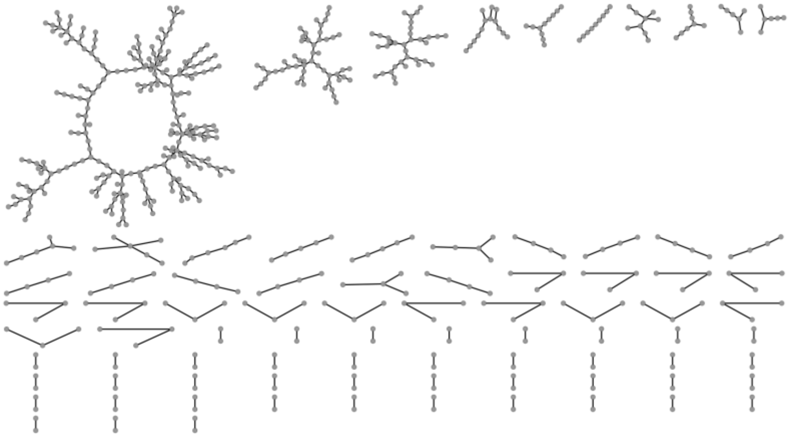

Figure A2. A visualization of a sample graph produced by the *Baseline* model.

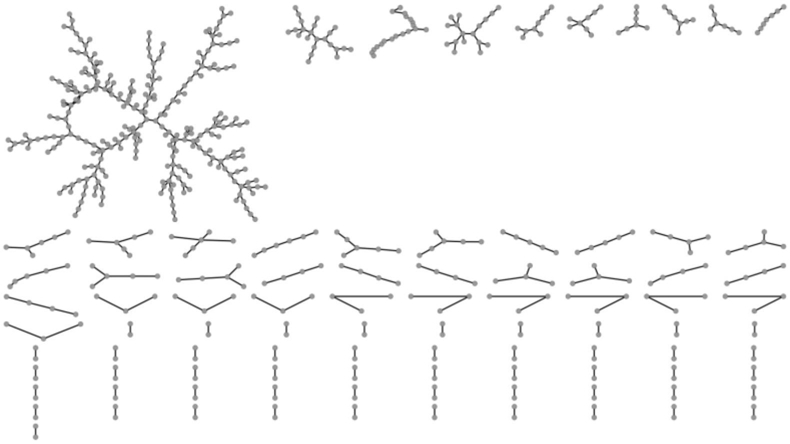

Figure A3. A visualization of a sample graph produced by the *Baseline* model.

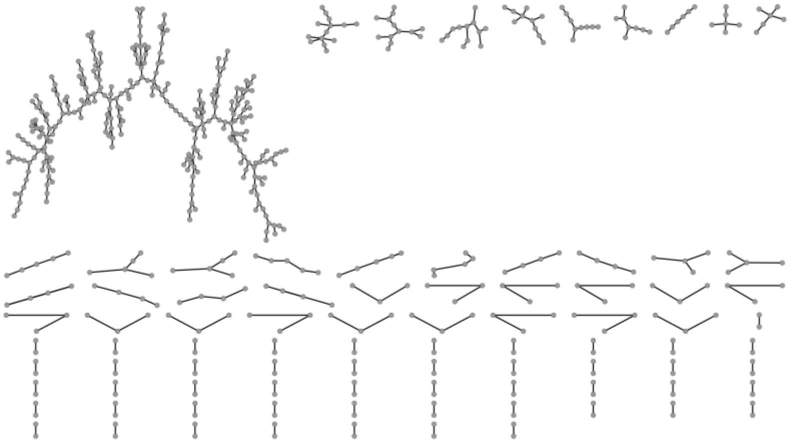

Figure A4. A visualization of a sample graph produced by the *Baseline* model.

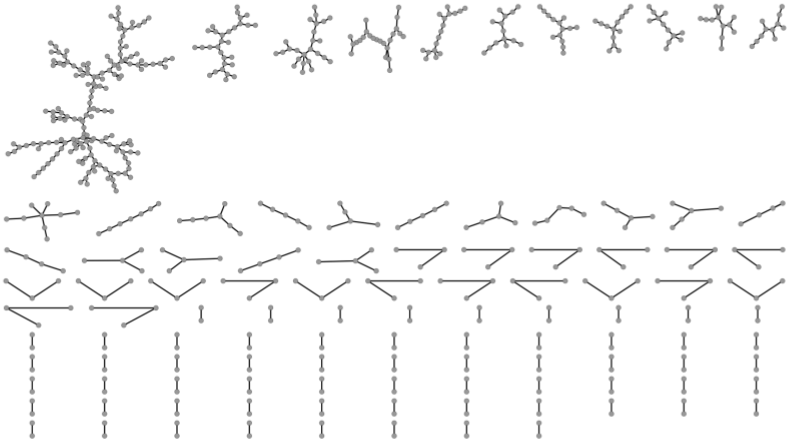

Figure A5. A visualization of a sample graph produced by the *Random* model.

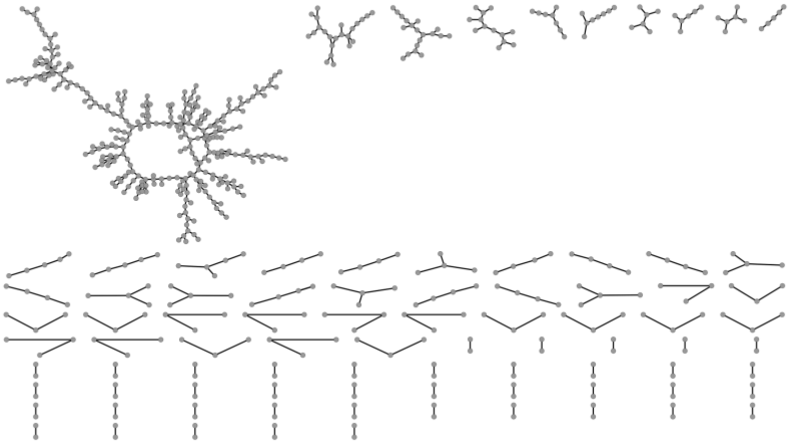

Figure A6. A visualization of a sample graph produced by the *Random* model.

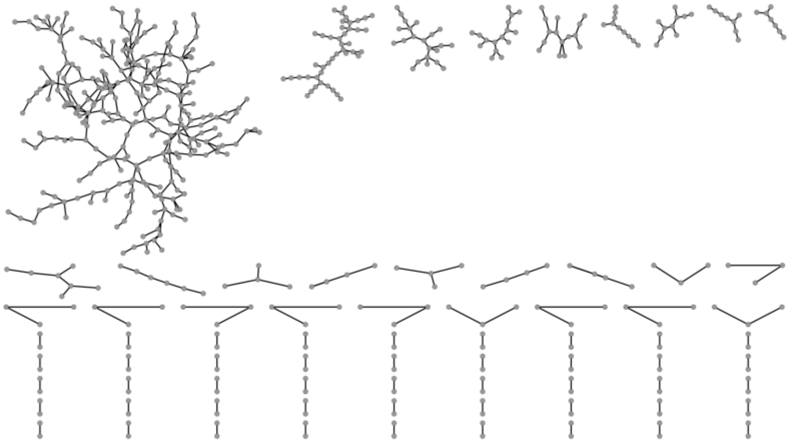

Figure A7. A visualization of a sample graph produced by the *Random* model.