Final Presentatiton

COMP702 Project of Modularity in empire network

Student name: Daozheng QU, Student ID: 201518453

MSc of Data science and AI, University of Liverpool

Supervisor: Michele Zito, Keith Dures.

Abstract:

What we design is an application in a professional field, that is, a combination of graph theory and modular theory and implemented by programming. We combine the known algorithms that are most suitable for this research (which will be explained in the summary), namely the Markov Chain algorithm and the Fast Newman algorithm, to generate a network(random planar graph) and transform it into another network(empire graph), and find its relatively optimal structural division and the value of the optimal structure division represented by modularity. We convert a random planar graph into multiple empire graphs with different numbers of vertices, and then perform modular processing, then run the experiment several times and take the average value to observe the value of modularity distribution. Finally, we set the vertex growth range and interval of the random planar graph to observe the modularity distribution of the random planar graph of different nodes and the difference in community division.

Acknowledgement:

I would like to sincerely thank the first supervisor for his help in the whole project and the meticulous answers to each question, such as random planar graph, empire graph, modularity problems, etc. I am also very grateful for the valuable comments provided by the second supervisor. I can correct many design errors and improve the details of the proof. I also thanks to Senior Sheng Chen for giving me inspiration on the traversal (single line of code) on the opposite side of the empire graph, which will enable me to better complete the transformation of the empire graph.

Ethical issues and data protection:

This project does not use any data of others, does not involve any ethical issues, and only uses publicly available resources for reference.It is strictly forbidden to use the project data and materials without the consent of the project supervisor and person in charge.

Structure of the presentation content (This content is divided into five parts):

1.SUMMARY OF PROPOSAL:

1.1.The aims of the project.

1.2.A summary of the design.

2.OUTPUTS:

2.1.A description of what was produced in the project.

2.2.Any interesting aspects of the implementation

3.EVALUATION:

3.1.An evaluation of what has been achieved.

3.2.Any shortfall from the original proposal.

3.3.Suggestions for future development.

4.CONCLUSION

5.REFERENCE

Main Text:

1.SUMMARY OF PROPOSAL:

1.1.The aims of the project.

Basic aim: Generate a small(50-100 nodes) random planar graph [1,2], and convert it into an empire graph according to the content of the empire graph conversion material given by the project, and find the empire graph whose outline structure is based on r=2 (25-50 vertices), and show its details about calculating the modularity value Q.

Challenge: Generate a random plan graph with 900-9000 nodes at an interval of 900 nodes, and repeat the above basic experiment 20 times. In the case of r = 2,3, 5, take the average Q value and analyze the distribution of the Q value.

1.2.A summary of the design.

We have made adjustments to some details of the original plan over time and actual needs, such as changing the number of runs of the experiment. According to the author's Newman's proof[3,9], the formula was revised in the design. We have not changed the algorithm selection. The algorithm selected by the original design is the most appropriate and accurate. The reason why we chose the original design algorithm will be shown below.

1.2.1. The design method and function of the program:

The design of basic aim: We generate a random planar graph according to the Markov chain algorithm, and then generate an empire graph (node merging) based on the data of the generated random planar graph. Then according to the modularization algorithm (Fast Newman), the generated empire graph is modularization, and the community is divided into the largest value of modularity. At this time, we have completed the design of program of basic aim, and we prove the reliability and feasibility of the design in the experiment.

The design of challenge aim: On the basis of the basic target program that we have completed, we designed a random planar graph of large nodes (continuously increasing the number of nodes) and the value of r converted into an empire graph is also different. Each random planar graph generates three empire graphs with different vertices, and modularization the empire graph. There is an empire graph with three vertices under the random planar graph of each large node, and the average modularity of each empire graph is taken after multiple runs.

1.2.2.Reasons for choosing the original design algorithm:

Random planar graph(Markov chain): We still use the Markov chain algorithm[5,6]. Firstly, the Markov chain algorithm performs a random memoryless walk in the state space when time is homogeneous. According to the algorithm of the Markov chain, all qualified (under the condition of keeping the planarity of the random planar graph) random planar sub-graphs can reach a uniform limit steady-state distribution, which is irreplaceable by other methods and is currently the best algorithm.

Modularity(Fast Newman): First of all, this algorithm was designed by the author Newman himself and proposed the concept of modularity for the first time. Second, it is very difficult for us to really get the real modularity in a large network, it is very time-consuming and laborious, and there is no good algorithm that can be directly obtained. Therefore, Newman proposed an idea based on greedy calculation, searching for all local optimizations until completion, and treating the results as global optimizations to obtain an approximate modularity. Compared with other algorithms (GN[9], Louvain method[10], etc.), this algorithm is faster and the accuracy almost same, so it is the best algorithm choice in this research

1.2.3.Pseudo-code of each algorithm

Random planar graph: initially G is the empty graph on vertices_num vertices, repeat iter_num times:choose positin (i,j), if G con remove it, else add {i,j}, if it keeps the graph planarity,and got a planar graph (vertices_nums).

Empire graph: Create n/r vertices of the empire graph,for each vertex v in planar graph:list all connections {v,u}add edge to empire graph between the empire of v and that of u.

Community detection and modularity:

```
define a class of Empire_Graph
define an inititial  function of G
#set a temporal empire graph for fast newman algorithm, which will
be changed during the calculation
# create zeros adjacency matrix
# node index in adjacency matrix
# Iterate each edge to create adjacency matrix
# find node i and node j index in adjacency matrix
# set adjacency matrix equal to one at [node_i_index, node_j_index]
# The community number of each node
# The partitions for empire graph
# every node will be a division
```

```
define a function for calculating delta Q
define a  function for calculating the value Q
define a  function Fast_Newman_Algo(empire_graph)
#set a division rotation
#set a max _q
#set a while loop(if division >1)
  set a Temp_Max_Delta_Q
  set a Max_Merge
  set an All_Delta_Q
  # Iterator all edges in temporal graph, repeatedly join communities
  together in pairs,
  # choosing at each step the join that results in the greatest increase
  (or smallest decrease) in Q.
```

```
  # Get communitiy number for node i and node j
  # Will not be considered, if node i and node j are in same
  community
 # calcualte delta Q before and after joining communitiy i and
  communitiy j
 #  Find maximal Delta Q
# If multiple delta Q are equal, join communities together in pairs with
maximal Q
# join communities together in pairs with maximal delta Q
# remove the edge after joining
# Find the devisions with maximal Q
#set G=create_empire(graph_ori, r)
#set the empire_graph = Empire_Graph(G)
# set division, max_Q = Fast_Newman_Algo(empire_graph)
```

2.1.A description of what was produced in the project.

We have completed the program design for the basic goals and the challenge-level goals and successfully run two programs. A small node application and a large node application. Due to time reasons, we only performed 40%(900-3600 nodes) of the total amount of the experiment, and produced 80 simulated graphs, 80 random planar graphs, 240 empire graphs, 240 community divisions and maximum Q values, and 12 average maximum Q values. The different number of nodes, different r values and average maximum Q value of the random planar graph are shown in the following figure.

nodes/r	average Q value of 20 times
900/2	0.442
1800/2	0.475
2700/2	0.491
3600/2	0.504

nodes/r	average Q value of 20 times
900/3	0.327
1800/3	0.355
2700/3	0.364
3600/3	0.372

nodes/r	average Q value of 20 times
900/5	0.231
1800/5	0.257
2700/5	0.266
3600/5	0.271

2.2. Any interesting aspects of the implementation.

With the increase in the number of random planar graph nodes, under a fixed number of simulations, the larger the Q value of the empire graph with a large number of nodes but the higher the density of its topology, sometimes it looks like a sphere. With the increase in the number of nodes in the random planar graph, under a fixed number of simulations, the trend of the number of small nodes is like a hospital tester that fluctuates within a certain level, and the large nodes seem to be sitting on a rocket close to a straight line.

3. EVALUATION:

3.1. An evaluation of what has been achieved.

Our basic design and the selected algorithm are feasible, effective, correct, which enables us to complete the program design of the basic goals and the more difficult goals, and run the program smoothly. However, due to time reasons, our relatively difficult program only yielded 40% of the results, but even this can reflect the conclusions and assumptions of this project. Generally speaking, the project has been completed, but it takes time to run more results. It is expected to continue to be described in detail in the dissertation.

3.2. Any shortfall from the original proposal.

It is no design alternative case, and many problems are not well mastered.Many algorithm problems and details have not been sorted out in place, and the basic materials and extracurricular materials are insufficient.The time management does not take into account that the time required for random planar graph and modularization increases sharply with the increase of the number of nodes, which results in the experiment with a larger number of nodes that has not been completed.

3.3. Suggestions for future development.

3.3.1. I hope to complete more random plane graph experiments with large numbers of nodes, and convert into more empire graphs under different r values. And explore the limit value of r, and finally observe the distribution of modularity of random planar graphs with different numbers of nodes and empire graphs with different vertices numbers, and get more experimental data.

3.3.2. I hope that with the development of hardware and the introduction of new algorithms in the near future, we can directly perform global optimization on large networks without resorting to the idea of greedy algorithms to complete all local optimizations to simulate global optimization[7,8].

4.CONCLUSION:

Based on the results of the two programs that have been completed, we analyzed and summarized the following conclusions. With the increase of the number of nodes in the random planar graph, the longer the number of iterations is required, the time to generate the random planar graph also increases exponentially.The number of iterations must be greater than the number of vertices for the program to run, and the number of iterations must be multiples of the number of vertices to have a good structure. Thus, when random planar graph transformed into an empire graph, the empire graph can have a good topological structure, which makes its modularity higher and more community groups. As the r value increases, more topology is lost, and the modularity value also decreases. The value of modularity is positively correlated with the number of iterations to generate the random graph, the number of nodes in the random plane graph, and the value of r used to transform the empire graph.

5.REFERENCE:

1. Denise, Alain and Vasconcellos, Marcio and Welsh, Dominic. (1996). 'The Random Planar Graph', Congressus Numerantium, 113, 61-79.

2. Bollobás, Béla (2001). Random Graphs (2nd ed.). Cambridge University Press.

3. M. E. J. Newman.(2004)'Fast algorithm for detecting community structure in networks', Phys. Rev. E, (69).066133.

4 .M. E. J.Newman.(2006). 'Modularity and community structure in networks'. Proceedings of the National Academy of Sciences of the United States of America. 103 (23): 8577–8696.

5. Dodge, Y. (2003) The Oxford Dictionary of Statistical Terms, OUP.

6. Gagniuc, Paul A. (2017). Markov Chains: From Theory to Implementation and Experimentation. USA, NJ: John Wiley & Sons, 159–163.

7. Jan A. Snyman (2005). Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing. ISBN 0-387-24348-8

8. M. E. J.Newman (2003), The structure and function of complex networks. SIAM Review 45, 167–256.

9. M.Girvan and M. E. J. Newman(2002), Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 7821–7826 .

10. Blondel, Vincent D; Guillaume, Jean-Loup; Lambiotte, Renaud; Lefebvre, Etienne (9 October 2008). "Fast unfolding of communities in large networks". Journal of Statistical Mechanics: Theory and Experiment. 2008 (10): P10008.