2018

CS310 - Project Specification

NORM EMERGENCE WITH FORGIVENESS

JONATHAN GOH (1602669)

Introduction

Social norms are patterns of behaviour that exist within a system where there is no central authority. The reason for a system such as this to exist range from privacy requirements to the efficiency of a system. As a result, these norms are needed for the individual entities to communicate with each other in a cooperative manner. These patterns are promoted through a mechanism called metanorms which dictates that one must punish those that do not follow the norm. The aim of this project is to investigate the effectiveness of metanorms in different topological systems and to introduce a forgiveness mechanism to the model.

Whilst Axelrod's model [2] shows how norms can be established within a population, it assumes that the network is complete, with all entities able to observe all other entities in the network. However, in real world applications, this may not always be the case and clusters of nodes may appear instead, with fewer connections to other clusters throughout the network as well as each node having much fewer connections. Systems like these can be seen everywhere, for example a university may have many nodes in its network, but not all students will have a connection with everyone else in that network. Instead, the students will form clusters of groups due to the characteristics of each person such as attending the same course or being a part of the same societies, and each cluster of students will have connections to other clusters in the network randomly due to other characteristics.

The project will also introduce a new forgiveness model whereby observers that see a node that does not comply with the norm in the system will also be given an option to forgive the act. In the real world, this mechanism may reflect someone who has committed an act of defiance but is not severe enough to receive a harsh punishment from an observer. Within the project, the way this interaction works has yet to be confirmed, however the general idea is that the node that doesn't comply ends up paying a small to no cost for that interaction.

Objectives

The aim of this project is to investigate the effect of forgiveness within different network topologies. The first part of the project involves the replication of the models generated in Mahmoud, Griffiths, Keppens and Luck's work [1] and once that has been achieved, to add another forgiveness mechanism to their model. Requirements for the project are as follows.

- Generation of networks with different topologies.
- Each node has a value for *boldness* and *vengefulness*
    - This value is how likely the node is to defy the norm and how likely the node is to punish a node that has not followed the norm
    - Values will change over the iterations of the model
- Nodes must be able to interact with each other properly.
    - Each entity must choose whether to comply to the norm or not.
    - If a node does not comply to the norm, then all observing entities will have a choice whether to punish that node with large cost to the defect and little cost to itself.
    - An observer that does not punish the node not complying to the norm may potentially receive a punishment themselves (metapunishment) from a second observer observing the situation.
- Output of the networks in graphical form such that model the characteristics of the nodes/network over time can be viewed clearly.

Methods

For this project, I have decided to use a mixture of plan-driven and agile methodologies. Whilst I have created a plan and a timetable for the progress of the project over the many weeks ahead, there is a very high chance that my plans and requirements may change over the course of the project, meaning I will have to adapt my project timeline to fit accordingly. A plan driven approach to this project is also useful since there are hard deadlines I need to meet to submit the various deliverables and having a structure for the project will greatly help me meet these goals.

Timetable

Time table has been outlined on the last page. Since I have much more time within my second semester (2 modules running in the second term as opposed to 4 for the first) I have scheduled most of the work for my project in that semester.

Meetings with my supervisor will run every week where we will discuss the progress and direction of the project.

Resources

The following are a list of resources that will be used within the project:

- Python – The language the program will be written in
- Git/GitHub – Create and store backups of my work so that the code can be accessed from different computers as well as access earlier versions of the code in case something happens to the current version
- NetworkX/graph-tool – Creating networks to use for the models
- Plotly(?) – Output of networks graphs to more clearly see the interactions between entities.

Risks

The following are a list of risks that may be encountered during the project.

- Something happening to the current version of the code such as losing the code or files getting corrupted. To prevent this from having a big effect on the project, I will make frequent backups on GitHub
- A python package that is being used malfunctioning with the code. This could be due to an update for the package that results in compatibility issues, or a server going down for an API in use. To remedy this, python packages that are needed will be downloaded and stored locally so that if there is an update that breaks the program, then going back to an earlier working version of the package may be the solution.
- As the networks become more complicated and the interactions between the nodes become more complex, the program may become much more difficult to manage and possible even run slowly, hindering progress. The best way to deal with this is to lay out my work clearly and to properly document the project throughout, as well as maintaining proper coding principles.
- The packages used may not support the networks or the network functions that are required for this project. If this happens then I will look for another package that meets the requirements or, in the worst-case scenario, create these network functions myself.

Legal, Social and Ethical considerations

Since the project does not involve the use of any personal data, there are no legal, ethical or social considerations that are relevant.

References

1. Samhar Mahmoud, Nathan Griffiths, Jeroen Keppens, Michael Luck, *Establishing Norms with Metanorms over Interaction Topologies*, available at https://www.dcs.warwick.ac.uk/~nathan/resources/Publications/jaamas-2017-author-version.pdf Accessed 10th October 2018.
2. Axelrod, R. (1986). An Evolutionary Approach to Norms. American Political Science Review, 80(4), 1095-1111. doi:10.1017/S0003055400185016

Project Planner

Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 2

Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Specification	1	2	1	2	100%
Progress Report	6	3	6	3	0%
Presentation Preparation	16	3	16	3	0%
Final Report	3	19	3	19	0%
Researching Axelrod's model	1	5	1	5	40%
Network Generation	3	4	3	4	0%
Creating UI	5	3	5	3	0%
Simulating Axelrod's model	6	4	6	4	0%
Testing different topologies	9	4	9	4	0%
Introducing Forgiveness	12	4	12	4	0%
Testing	16	3	16	3	0%
Overflow time	19	3	19	3	0%