CS166 Assignment 3
Network simulation

In this assignment, you improve the social dynamics model we covered in Session 7.2 on Adaptive Networks. You propose your own changes to the model to make the dynamics more realistic in some way, implement your improvements in a simulation, and analyze the expected (theoretical) and actual (experimental) effects of your changes on the model.

Part 0: The basic model

- Review the model we discussed during breakouts in Session 7.2.
- You can find the description of the model in the pre-class work of that session.
- You can find an implementation of the model in the Activity 2 notebook used during breakouts. There is also a copy here.
- This model forms the basis of your assignment, and you should feel free to reuse and modify the code from the notebook for your own implementation.

Part 1: Propose 2–3 modifications

Choose a few ways to modify the existing rules of the social dynamics model. For some inspiration, look at the ideas below. You can use one or more of these ideas, along with any of your own devising.

If you are uncertain about whether your modifications are right for this assignment, please come and discuss them during office hours or after class early on so that you have time to analyze and implement them.

Ideas:

- **Multiple topics** Each person has an opinion on multiple topics, so rather than having a single value between 0 and 1 for each opinion, have a vector (array) of values where each entry in the vector represents a person's opinion on one topic. Modify the rules of the model so that people have a conversation about a particular topic (selected randomly, or selected based on some criteria that you have to design) and update their opinions and relationship strength based on that conversation.
- **New relationships** Implement a different mechanism for creating new relationships – not entirely random, like in the basic model, and not with a weight of 0.5. You should determine a realistic process for how people who haven't met get introduced and how the new relationship can strengthen or weaken over time.
- **Persuasiveness** Some people are more charismatic or convincing than others, and can, therefore, influence opinions and relationship strengths more than others. Include a persuasiveness parameter for each person in the model and update the model rules to incorporate this dynamic.

Part 2: Local analysis

- Analyze how your proposed changes affect the relationship between 2 people. You can think of this as running the simulation on a network with 2 nodes and 1 edge.
- Analyze mathematically for which values of your parameters you should expect opinions to converge or diverge and relationship strengths to increase or decrease.
- Use this analysis to select appropriate values to use when running your simulation (Part 3). You could also select a reasonable range of values for each parameter rather than specific numerical values.
- Use one or more vector field plots to demonstrate for which opinion and relationship strength values you expect clusters to form or to split apart in the simulation. Refer to the readings and pre-class work of Lesson 7.2 on Adaptive Networks for information on making vector field plots.

Part 3: Implementation

- Choose the type of graph to use – select and motivate for one of the types of random graphs you have encountered in this course.
- Modify the Python code from the basic model, or write your own version of the code from scratch, to implement your modified model.
- Make sure you structure and document your code properly so that it is easy to read and understand.

Part 4: Simulation analysis

- Run experiments using your simulation to determine under what conditions social clusters form and under what conditions they split apart.
- Report on whether and how your simulations show patterns observed in real life.

Deliverables

Report

- Typeset your work and submit your report as a PDF file.
- For your modified model (Part 1 above), describe
    - your assumptions,
    - your update rules, and
    - all parameters along with their meanings and any numerical constraints (for example, must be greater than 0).
- Motivate why your modified model is an improvement on the base model by referring to the real-world situation(s) it is trying to capture.
- Report on the results of your local analysis (Part 2 above). Include figures, plots, mathematical derivations, and explanations as needed to describe and explain what the local analysis tells you about how the overall model will behave and what realistic parameter settings are.
- Report on your choice of random graph for running simulations. Motivate why you chose a particular type of graph (or graphs if you tried out more than one type).
- Report on the results of your simulations (Part 4 above).

Implementation

- Write your code in a Jupyter / CoCalc notebook, also containing any output (text or plots) generated by running the code. Attach your notebook to your assignment submission as a .ipynb or zipped .ipynb file.
- Write Python code to implement a simulation of your modified model, as described in your report.
- Make sure your code is neat, well-structured, and documented.
- Make sure your code runs by selecting "Kernel | Restart and run all" from the notebook menu.