Analysis of Social Network Simulation

Reece Jones

Notation

number of people in a network.

probability of a person liking an interactable post.

probability of a person following the creator of a liked post.

person , where .

set of people person is following.

set of posts liked by person .

set of posts made by person .

any post.

set of interactable posts for person .

Simulation model

The simulation is based on a social media network consisting of people and posts made by those people. Each person may *follow* zero or more other people and may *like* zero or more posts, forming connections within the network. The simulation models the evolution of these connections through time. The flow of time in the network model is quantised into *timesteps*. At each timestep, the evolution algorithm is applied, producing a (typically) new network structure.

The algorithm iterates the people in the network and possibly causes them to form new connections to other people and posts. For each person, a set of “interactable” posts is generated, consisting of posts made and liked by people that person is following. This emulates the behaviour of a real social media application, in which one typically is shown a list of posts/activities from connected people.  
For each interactable post, there is a chance for the person to like the post, and if liked, a chance for the person to follow the creator of the post. (An additional like chance scaling factor may be present for a post, however this will be ignored in this analysis for simplicity.) These probabilities are constant parameters of the simulation. The like and follow chances are notated and , respectively.

The simulation is considered complete when each person likes every post they possibly can and follows every person they possibly can. Note that this is dependent on the initial network structure and is not the same as a fully connected graph; see the code and documentation for details.

Computation complexity of simulation

The simulation code has been developed and optimised for performance. The time complexity of basic network operations are as follows:

* Iterating all people:
* Iterating :
* Iterating :
* Iterating :
* Checking if :
* Checking if :
* Adding to :
* Adding to :

(A complete explanation of the code is available in the documentation, from which the time complexities may be derived.)

Practical performance case 1: linear network

To begin understanding the behaviour of the simulation, we will first examine a very basic network structure that resembles a linear graph, described by the following:

for

Such an example is unlikely to occur in reality; however, it is simple to analyse.