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.

set of all people in a network.

person , where .

set of creators of posts .

set of people person is following.

set of posts liked by person .

set of posts made by person .

set of interactable posts for person .

network parameter after the th timestep, where .  
 represents the state before simulation begins.  
 represents the state after the simulation is complete.

random subset of , where each element has probability of being included.

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 specific post, however this will be ignored in this analysis for simplicity.) These probabilities are constant parameters of the simulation.  
The simulation can be represented mathematically as:

The simulation is considered complete when each person likes every post they possibly can and follows every person they possibly can via simulation alone. 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.

Computational complexity of simulation

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

* Iterating :
* Iterating :
* Iterating :
* Iterating :
* For a person , checking :
* For a post , checking :
* Adding a person to :
* Adding a post to :
* Finding the creator of a post:

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

Then the time complexity for evaluating timestep is:

That is, proportional to the number of total number of post interactions across all people. However, the value of is non-trivial and heavily dependent on the network structure. The following sections will derive solutions for certain structures.

Randomised network structure

In this network, each person follows a random number of randomly selected people and has a random number of posts. The number of follows and posts per person are normally distributed with . People to be followed are chosen by simple random sample (without replacement).

We additionally assume the number of follows and post likes for any person at some timestep is uniformly distributed with:

Then it follows that

Hence the time complexity of timestep will be .

At each timestep, the random samples from and may include people already followed and posts already liked

Linear network structure

In this network, the set of follows resembles a linear graph, with the last person having one post. More precisely:

for

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