Markov chain model representation of information diffusion in social networks

LOUISE A. DENNIS, *Department of Computer Science, University of Manchester, Manchester, UK, M13 9PL.*
E-mail: louise.dennis@manchester.ac.uk

YU FU, *Department of Computer Science, University of Liverpool, Liverpool, UK, L69 3BX.*

MARIJA SLAVKOVIK, *Department of Information Science and Media Studies, University of Bergen, Bergen, Norway, 5020 Bergen.*

Abstract

The spread of information in a social network has received renewed interest as social media becomes an increasingly popular channel of communication. We are interested in the phenomenon of social diffusion of a piece of information in the presence of a contradicting information in the network. Specifically we explore the use of formal methods for verification in studying this phenomena. Using Monte Carlo simulation and the probabilistic model checker (PRISM) we are able to represent social networks and confirm an earlier conjecture that disseminating new information rapidly is resistant to the presence of contradicting information.

1 Introduction

Social network analysis is concerned with the structures of social relations and the graph they form, as well as how that structure influences, and is influenced by the spread of information through the networks (14; 28).

Information Diffusion is the process by which information spreads through a network. Social networks are naturally modelled as graphs of agents, in which the agents are represented by vertices in the graph which are connected by edges if the agents can share/communicate information. Diffusion has been extensively studied in the social network analysis literature; see, e.g. (15) and (21) for an overview. In particular, the impact of the social network graph on the diffusion process has been studied. Social networks of communication have physically changed over the past 30 years. In particular, aspects of these networks, such as the distance between two nodes and the speed of communication, have changed drastically. This observation has revived interest in the study of information diffusion, including work that represents the phenomena using Markov chains (e.g. (4)) as we do here.

The rise of social media has also generated interest in the informational and motivational states of the agents in the network—such as their receptiveness to opposing views and motivations for sharing or not sharing information—not just the relations between agents (13; 22; 26; 29).

Formal verification involves proving or disproving that a system is compliant with a formally specified property (11). Arguably the most practical method of formal verification is model-checking (7), in which all possible executions of a system can be examined automatically based

Vol. 32, No. 6, © The Author(s) 2022. Published by Oxford University Press.

Advance Access Publication on 11 March 2022 https://doi.org/10.1093/logcom/exac018

on a model of the system. However, if exhaustive exploration of the model proves intractable then the same model may be used in a simulation based approach.

In previous work (8), which we extend and build on here, we have built formal specifications of social networks and diffusion properties using the input language of a probabilistic model checker (PRISM). Unfortunately, even simple models that take account of both network structure and an agent's informational state proved largely intractable for model checking on networks of any significant size. We therefore supplement the use of PRISM with Monte Carlo simulation, which samples the space of information diffusion behaviour. This contrasts to PRISM's exhaustive exploration of all possible outcomes. Using Monte Carlo simulation we were able to analyse significantly larger networks and validate a tentative result from (8) about the beneficial effect of disseminating new information rapidly.

2 Background

Several models of information diffusion through influence have been proposed, although the task of finding a good model remains challenging (6). The social influence models used to define processes of diffusion can broadly be classified into two classes: infection models and threshold models, with the possible exception of the recent Simmelian model (26). Infection models have their roots in early research in social networks from epidemiology. In our context the concept of 'infection' translates to a concept of 'influence'—an agent in a social network is infected by some piece of information if they adopt or are influenced by it. In threshold models, an agent is influenced when the number of her influenced neighbours passes a certain threshold (31). We concentrate here on the infection model, first presented in (8), though note that that paper also contains a Markov chain formalisation of a threshold model, which we do not consider.

2.1 Infection models

One of the classic infection models is the SIS model (2). In this infection model each of the nodes in the graph can be in one of two states: infected or susceptible to infection. At time t, $s(t)$ represents the susceptible proportion of the total population N, $i(t)$ represents the infected proportion, and λ represents the daily contact rate, which means the proportion of the susceptible users infected by infected users in the total population, where $s(t) + i(t) = 1$. At time $t = 0$ the proportion of infected nodes is i_0. The SIS model assumes that μ represents the daily rates of the 'cured' nodes (a node can now become uninfected). The SIS model can be described by

$$\begin{aligned} &\tfrac{di}{dt} = \lambda i(1-i) - \mu i \\ &i(0) = i_0. \end{aligned}$$

The SIS model is not suitable for our purposes since it has no account of the internal states of the agents, nor does it represent the graph topology. What we want is a representation in which both the agent's attitude to information and their position within the graph are relevant.

2.2 Discrete-time Markov chains

Markov chains are an elegant formalisation of probabilistic processes that transition between states. A *discrete-time Markov chain (DTMC)* is one in which a discrete model of time is assumed with states transitioning at set time points. Formally:

Definition 1
(18) (Discrete-time Markov chain). A DTMC is a tuple $D = (S, s_0, P, L)$, where S is a finite set of states, $s_0 \in S$ is a distinguished initial state, $P : S \times S \rightarrow [0, 1]$ is a transition probability matrix such that $\sum_{s' \in S} P(s, s') = 1$ for all $s \in S$, and $L(s) \subseteq AP$ is a labelling with atomic propositions from some fixed set AP.

A DTMC describes a set of execution paths through the state space S where P gives the probability of one state moving to the next and L describes propositions that are true in any given state.

Definition 2
(Reward Function) We can specify a *reward function*, $\rho : S \rightarrow \mathbb{R}$, which assigns some reward value to the states in S.

The use of a reward function on states allows us to investigate the expected reward at some time step, t, in the system. This proves a particularly powerful tool in the study of information diffusion where we model the number of agents who have adopted some idea in a particular state as the reward.

The role of the labelling function L in the DTMC enables us to identify properties of states that may be used in reasoning with the DTMC—for instance, a reward function might be defined in terms of the labels on states. We do not utilise state labels in this paper.

We will use DTMCs to represent both individual agents within a network and a social network as a whole. To enable this we utilise ideas of synchronisation and parallel composition drawn from Markov decision processes (MDPs). MDPs are similar to Markov chains but incorporate non-deterministic as well as probabilistic transitions. We will therefore briefly cover MDPs, how a DTMC can be induced from an MDP and adapt the definition of parallel composition of MDPs to DTMCs. Our presentation follows that in (12).

Definition 3
(Markov Decision Process) An MDP is a tuple $\mathcal{M} = (S, s_0, \alpha_{\mathcal{M}}, \delta_{\mathcal{M}}, L)$ where S is a *finite* set of states, $s_0 \in S$ is an initial state, $\alpha_{\mathcal{M}}$ is a *finite* alphabet and $\delta_{\mathcal{M}} : S \times \alpha_{\mathcal{M}} \rightarrow Dist(a)$ is a (partial) probabilistic transition function. The set of available actions in a state, s, is given by $A(s) \overset{\text{def}}{=} \{a \in \alpha_{\mathcal{M}} \mid \delta_{\mathcal{M}}(s, a) \text{ is defined}\}$. The function $L : S \rightarrow 2^{AP}$ is a labelling function mapping each state to a set of atomic propositions taken from a set AP.

Transitions between states in an MDP, $\mathcal{M}$, occur in two steps. First a choice between one or more available *actions* from the alphabet $\alpha_{\mathcal{M}}$ is made. To prevent deadlocks, we assume that $A(s)$ is non-empty for all $s \in S$. The selection of an action $act \in A(s)$ is nondeterministic (i.e., the MDP makes no assumptions about the probability of any action being selected). Secondly, a successor state s' is chosen randomly, according to the probability distribution $\delta_{\mathcal{M}}(s, act)$, i.e. the probability that a transition to s' occurs equals $\delta_{\mathcal{M}}(s, act)(s')$.

The concept of selecting an action that is available in state s, performing the action and then calculating a probabilistic outcome for the result of the action, is useful for formalising the behaviour of social networks where the actions can be considered the transmission of messages on the network. While we could use an MDP for this, we simplify matters by assuming that there is an equal probability of selecting an action act from the set of available actions in a state s, i.e. if $\sigma : S \rightarrow Dist(\alpha_{\mathcal{M}})$ represents the probability distribution for selection of an action act in some state s then $\sigma(s)(act) = \frac{1}{|A(s)|}$ if $act \in A(s)$ and 0 otherwise.

DEFINITION 4

An *induced DTMC* is a tuple $(S, s_0, \alpha_{\mathcal{M}}, \delta_{\mathcal{M}}, L)$ where S, s_0 and L are as for an MDP, $A(s) \neq \emptyset$ for all $s \in S$, and

$$P(s, s') = \sum_{act \in A(s)} \frac{\delta_{\mathcal{M}}(s, act)(s')}{|A(s)|}.$$

We use PRISM style notation to represent induced DTMCs. In this notation for each action, *act*, and each state s in which *act* is available we write a transition of the form:

$$s \xrightarrow{act} p_1 : s_1 \wedge \ldots \wedge p_n : s_n,$$

where $p_i = \delta(s, act)(s_i)$. We can represent an induced DTMC as a set of such transitions.

We will used these induced DTMCs to represent individual agents within a social network and their *parallel composition* to represent the network as whole. The following definition of parallel composition is from the definition for parallel composition of MDPs in (12).

DEFINITION 5

(Parallel composition of induced DTMCs). If $\mathcal{M}^i$ are induced DTMCs $(S^i, s_0^i, \alpha_{\mathcal{M}}, \delta_{\mathcal{M}^i}, L^i)$, which share the same alphabet of actions, $\alpha_{\mathcal{M}}$, for $i = 1, 2$ then their *parallel composition*, denoted $\mathcal{M}^1 \parallel \mathcal{M}^2$ is given by the DTMC induced from $(S^1 \times S^2, (s_0^1, s_0^2), \alpha_{\mathcal{M}}^1 \cup \alpha_{\mathcal{M}}^2, \delta_{\mathcal{M}^1 \parallel \mathcal{M}^2}, L)$ where $\delta_{\mathcal{M}^1 \parallel \mathcal{M}^2}$ is defined such that $\delta_{\mathcal{M}^1 \parallel \mathcal{M}^2}((s^1, s^2), act) = \delta_{\mathcal{M}^1}(s^1, act) \times \delta_{\mathcal{M}^2}(s^2, act)$ and $L(s^1, s^2) = L^1(s^1) \cup L^2(s^2)$. This is a DTMC so long as $A(s^1) \cap A(s^2) \neq \emptyset$ for all (s^1, s^2) reachable from (s_0^1, s_0^2).

Definition 5 allows us to represent a social network by representing the agents individually and the effect they have on each other as *synchronised actions* (that is each agent makes a transition labelled by the same action *act*). The probability of a global transition from one state to a next state therefore depends upon the existence of transitions with the same action in each of the agent DTMCs. Synchronised actions thus allow us to represent the effect the actions of one agent have on other agents in the network.

Transitions between states in the induced DTMC, occur in three steps. First we pick an action *act* that is available in s at random with an equal chance of any available action being selected. We then calculate the next local state for each agent i using the probability distribution defined by $s^i \xrightarrow{act} p_1 : s_1^a \wedge \ldots \wedge p_n : s_n^a$. We then compose these these local states into the next global state.

We can use repeated parallel composition to create an induced DTMC for an arbitrary numbers of agents.

2.3 PRISM

PRISM (19) is a probabilistic symbolic model checker in continuous development since 1999, primarily at the Universities of Birmingham and Oxford. Typically a model of a system is supplied to PRISM in the form of a probabilistic automata. This can then be exhaustively checked against a property written in PRISM's own probabilistic property specification language, which subsumes several well-known probabilistic logics. PRISM has been used to formally verify a variety of systems in which reliability and uncertainty play a role, including communication protocols and biological systems (10; 20).

In our models we use DTMC with a reward function as our probabilistic automata.

2.4 Monte Carlo simulation

Monte Carlo simulation is a broad term for a range of techniques for using random sampling and statistical modeling to mimic the operations of complex systems. If a system is represented as a DTMC then there is a natural application of Monte Carlo simulation to the model in which each simulation run begins in the starting state of the model and then selects the next state according to the probability distribution over transitions.

It should be noted that model checkers such as PRISM can be used as simulation systems in this way; however, PRISM was unable to construct the models necessary to allow this sampling so we opted instead to write our own simulator described in Section 5.

3 A Markov Chain Infection Model of Information Diffusion in Social Networks

As has been noted in the literature (5) the transmission of information around a social network may depend both on the features of the specific agents in the network and on the structure of the social network itself. We are interested in how network structure affects the spread of ideas. Further, we want to see how agent properties contribute to the global effect.

As an example of an agent feature that might influence contagion we consider how one idea may be associated with an 'anti-idea, that might either cause an idea to be abandoned (analogous to recovering from infection in traditional model) or might cause other behaviour (e.g. greater adherence to the original idea, modifications to network structure and so on). Taking this example, which to the best of our knowledge has not been considered in social network analysis, is motivated by the insight from psychology that 'once formed impressions are remarkably perseverant' (23). In this case we use the current informational state of the agent to inform how likely it is to adopt an opinion. Once adopted it will broadcast the opinion to its network.

We present a class infection models represented as the parallel composition of induced DTMCs, generated from a social network graph, the initial informational states of the agents and two probabilities. In an infection model, each agent a_i in the network can be in one of three states. Either the agent agrees with some idea ϕ (written as state $s_{\phi}^{a_i}$) or it disagrees with the idea ($s_{\neg\phi}^{a_i}$) or it is indifferent to ϕ (written as $s_{\perp}^{a_i}$)). If there are n agents in the network, there are 3^n states in the global model.

An agent may broadcast a message in favour of ϕ (respectively, $\neg\phi$) to all of its connections if it agrees with ϕ. We treat this as an action $a_i\_says_{\phi}$ (resp. $a_i\_says_{\neg\phi}$), which is available in state $s_{\phi}^{a_i}$ (resp. $s_{\neg\phi}^{a_i}$).

On receiving a message in favour of ϕ (resp. $\neg\phi$) there is a probability λ that the agent will adopt the idea ϕ (resp. $\neg\phi$) if it is currently indifferent to it and a probability μ that it will abandon the idea $\neg\phi$ (resp. ϕ) if that is already held. Receiving a message is modelled as a transition labelled with the same action as sending the message.

Note that the definition of a transition between two global states the infection model represents the sending of a message by one agent synchronised with the receiving of the message by its connected agents. Thus, implicitly, our model selects an agent, at random, to transmit its belief about ϕ and all agents receiving this message then update their state accordingly.

Definition 6

A *social network infection model*, SN, is a tuple, $\langle MD, G\rangle$. MD represents the parallel composition of n induced DTMCs, a_i, each representing an agent in a social network, and a graph G in which each

$$
\begin{array}{llr}
s^{a_i}_{\phi} \xrightarrow{a_i\_says_{\phi}} & 1 : s^{a_i}_{\phi} & \qquad (1)\\
s^{a_i}_{\phi} \xrightarrow{a_j\_says_{\phi}} & 1 : s^{a_i}_{\phi} \qquad \text{if } i \neq j & (2)\\
s^{a_i}_{\phi} \xrightarrow{a_j\_says_{\neg\phi}} & \mu : s^{a_i}_{\neg\phi} \wedge (1-\mu) : s^{a_i}_{\phi} \qquad \text{if } i \neq j \wedge cn(i,j) & (3)\\
s^{a_i}_{\phi} \xrightarrow{a_j\_says_{\neg\phi}} & 1 : s^{a_i}_{\phi} \qquad \text{if } i \neq j \wedge \neg cn(i,j) & (4)\\
 & & (5)\\
s^{a_i}_{\neg\phi} \xrightarrow{a_i\_says_{\neg\phi}} & 1 : s^{a_i}_{\neg\phi} & (6)\\
s^{a_i}_{\neg\phi} \xrightarrow{a_j\_says_{\neg\phi}} & 1 : s^{a_i}_{\neg\phi} \qquad \text{if } i \neq j & (7)\\
s^{a_i}_{\neg\phi} \xrightarrow{a_j\_says_{\phi}} & \mu : s^{a_i}_{\phi} \wedge (1-\mu) : s^{a_i}_{\neg\phi} \qquad \text{if } i \neq j \wedge cn(i,j) & (8)\\
s^{a_i}_{\neg\phi} \xrightarrow{a_j\_says_{\phi}} & 1 : s^{a_i}_{\neg\phi} \qquad \text{if } i \neq j \wedge \neg cn(i,j) & (9)\\
 & & (10)\\
s^{a_i}_{\perp} \xrightarrow{a_j\_says_{\phi}} & \lambda : s^{a_i}_{\phi} \wedge (1-\lambda) : s^{a_i}_{\perp} \qquad \text{if } i \neq j \wedge cn(i,j) & (11)\\
s^{a_i}_{\perp} \xrightarrow{a_j\_says_{\phi}} & 1 : s^{a_i}_{\perp} \qquad \text{if } \neg i \neq j \wedge cn(i,j) & (12)\\
s^{a_i}_{\perp} \xrightarrow{a_j\_says_{\neg\phi}} & \lambda : s^{a_i}_{\neg\phi} \wedge (1-\lambda) : s^{a_i}_{\perp} \qquad \text{if } i \neq j \wedge cn(i,j) & (13)\\
s^{a_i}_{\perp} \xrightarrow{a_j\_says_{\neg\phi}} & 1 : s^{a_i}_{\perp} \qquad \text{if } i \neq j \wedge \neg cn(i,j) & (14)
\end{array}
$$

FIGURE 1. Transition system for agent a_i in an infection model

agent, a_i forms a node in the graph and the edges of the graph represent social network connections. We write $cn(i,j)$ if agents a_i and a_j are connected by an edge in G, i.e. $cn(i,j) \overset{\text{def}}{=} \langle a_i, a_j \rangle \in edges(G)$.

Each agent (induced DTMC) in *MD* is defined by the transition system shown in Figure 1.

We note that it is straightforward to generate the parallel DTMC in a social network infection model automatically from the graph G, the initial state for each agent, and the probabilities λ and μ.

4 Model-Checking Infection Models

To start, we considered a fully connected network (FCN) of 10 agents. We seeded the network with one agent believing ϕ and one agent believing $\neg\phi$ (all other agents indifferent) and set the probabilities of infection, λ and μ to 0.5. We generated a social network infection model from the graph of this network for use with PRISM.

To create a reward function we first created a set of the number of agents in some state $s = \{s^{a_0}, \ldots, s^{a_n}\}$ that believed ϕ:

$$B_{\phi}(\{s^{a_0}, \ldots, s^{a_n}\}) = \{s^{a_i} | s^{a_i} \in \{s^{a_0}, \ldots, s^{a_n}\} \wedge s^{a_i} = s^{a_i}_{\phi}\}$$

Then defined $\rho(s)$ as proportion of the total number of agents in state $s^{a_i}_{\phi}$, i.e. the size of $B_{\phi}(s)$ divided by n:

$$\rho(s) = \frac{|B_{\phi}(s)|}{n}.$$

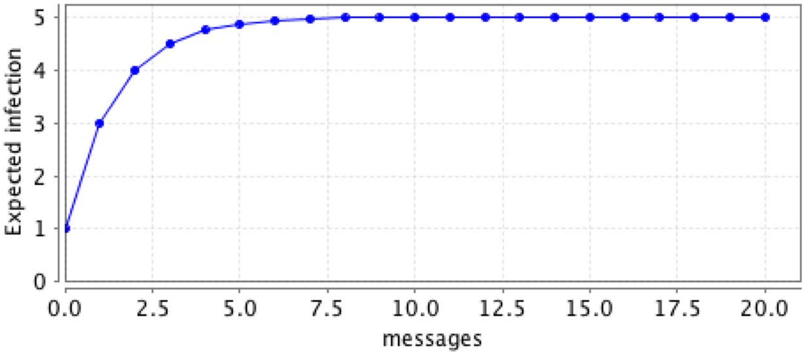

FIGURE 2. Expected number of infected agents per message sent on a FCN

We used PRISM to calculate the expected reward over time (represented by the number of actions taken/messages broadcast within the network).

Figure 2 shows that this network quickly converges to a state where the expectation is that half the agents believe ϕ—the expected reward is 5.

We are not very interested in FCNs. Research in the information diffusion under the SIS model from early on has shown that the structure of the network has a big influence on the effectiveness of the contagion (27) and fully connected models are not very realistic in terms of social networks.

We want to have a 'higher detail' insight into the impact a particular graph has on the spread of information. We generated a random network that satisfies the criteria for modelling a social network as a random graph as outlined in (24): the maximal degree of separation is low, the probability of an edge between two agents is higher if they have mutual neighbours, and the network has a skewed degree distribution. This network contained 10 agent nodes, some with a minimum of 2 connections within the network and one with 8 connections. We initially studied the spread of ideas within this network with $\lambda = \mu = 0.5$ and ϕ and $\neg\phi$ inserted in poorly connected agents (i.e. agents with only two connections within the network), well-connected agents (i.e. agents with 6 connections) and when the agent with idea ϕ had 8 connections while the agent with idea $\neg\phi$ had only 2 connections. We generated infection models for this network and the initial states we were interested in and used PRISM to calculate the expected reward over time/messages broadcast. The results are shown in Figure 3.

As it can be seen in the case where the initial agents have similar numbers of connections, the expected number of infected agents converges to 5 (converging more rapidly in the case where the initial agents have more connections). However, in the case where the agent initially wishing to disseminate ϕ has more connections than the agent wishing to disseminate $\neg\phi$ then the number of agents believing ϕ converges to just under 6—showing that the initial advantage had a long term effect. This result came as a surprise to us—our hypothesis was that over time the network would equalise to a steady state where roughly half the nodes held idea and half held the anti-idea. This was based on an assumption that simply selecting a seed node with many (or few) connections would have little long-term effect since those connections would themselves vary in connectivity and so the speed that an idea and anti-idea moved through the network would ultimately be the same and would be governed by the network shape as a whole and not by the starting conditions of two nodes. As far

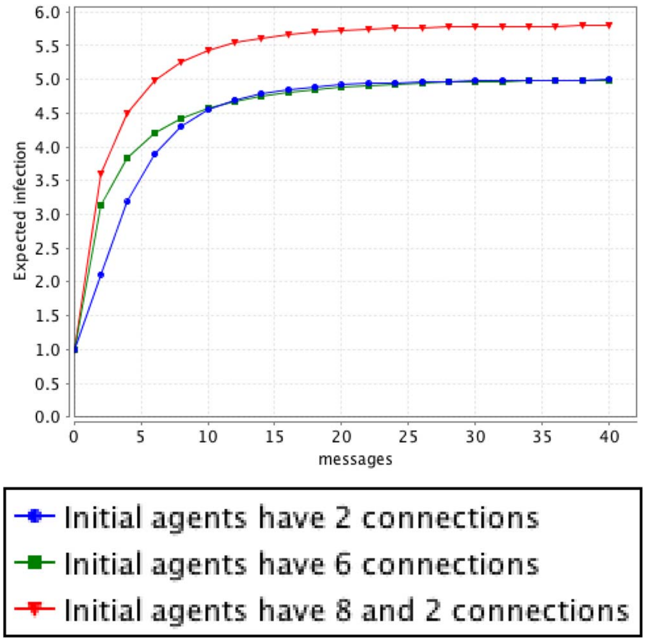

FIGURE 3. Expected number of infected agents per message sent on a randomly generated network

as we are aware this effect is not one that has been studied in the context of diffusion in the literature. We generated another 9 networks (for a total of 10) and observed the same effect in all of them. We further hypothesised that the effect was caused by the small size of the networks, thus allowing the initial advantage to quickly translate into a network-wide advantage before the disadvantaged ‘anti-idea’ had a chance to establish itself. However, we were unable to investigate whether the same effect held for larger network sizes using PRISM. In fact PRISM struggled to even build models for larger networks, let alone calculated expected rewards over time[1].

5 A Monte Carlo Simulation Tool for Influence Models

We implemented a simulation tool for our Markov chain influence models. This tool takes as input a graph, two seed nodes for an idea, ϕ and its corresponding anti-idea, $\neg\phi$, the probability that an agent will change its mind and the number of messages to be exchanged in a simulation. This algorithm is shown as Algorithm 1. It performs one run through the system and calculates, as a reward, the number of agents in the graph in agreement with the idea ϕ.

[1] The reasons for this continue to be opaque. While we would not expect PRISM to cope with networks consisting of thousands of nodes, a model with considerably more than 3^{10} states should not be an unreasonable challenge for the tool, particularly if used in ‘simulation’ mode rather than ‘model-checking’ mode.

Algorithm 1: Infection Module Simulation

```
Function reward(idea, anti_idea, graph, $\lambda$, $\mu$, messages)
    $edges \longleftarrow$ edges($graph$) ;
    $idea.state \longleftarrow$ agree ;
    $anti\_idea.state \longleftarrow$ disagree ;
    $Infected \longleftarrow [idea, anti\_idea]$;
    $m \longleftarrow 0$;
    while $m < messages$ do
        $node \longleftarrow$ random($Infected$) ;
        for $n \in$ neighbours(graph, node) do
            if $node.state \neq n.state$ then
                if $n.state$ = indifferent & random_float() $\leq \lambda$ then
                    push($n$, $Infected$) ;
                    $n.state \longleftarrow node.state$ ;
                else if random_float () $\leq \mu$ then
                    $n.state \longleftarrow node.state$ ;
        $m \longleftarrow m + 1$ ;
    $reward \longleftarrow 0$;
    for $n \in$ vertices(graph) do
        if $n.state = agree$ then
            $reward \longleftarrow reward + 1$;
    return $reward$ ;
```

Algorithm 2: Monte Carlo Simulation Algorithm for idea and anti-idea seed nodes with similar connectivity

```
Function similar(graph, $\lambda$, $\mu$, messages, runs)
    $nodes \longleftarrow |$ vertices($graph$) $|$;
    $node\_list \longleftarrow$ sort_by_connectivity(vectices($graphs$));
    $anti\_idea\_idx \longleftarrow$ random($nodes$);
    $idea\_idx \longleftarrow anti\_idea\_idx + nodes/10-$ random($nodes/10$) ;
    $reward \longleftarrow 0$ ;
    $i \longleftarrow 0$;
    while $i < runs$ do
        $reward \longleftarrow reward+$ reward($node\_list[idea\_idx]$,
          $node\_list[anti\_idea\_idx]$, $graph$, $\lambda$, $\mu$, $messages$)
    return $\frac{reward*100}{runs*nodes}$
```

Algorithm 3: Monte Carlo Simulation Algorithm idea and anti-idea seed nodes with dissimilar connectivity

```
Function dissimilar($graph$, $\lambda$, $\mu$, $messages$, $runs$)
    $nodes \longleftarrow |$ vertices($graph$) $|$;
    $node\_list \longleftarrow$ sort_by_connectivity(vectices($graphs$));
    $anti\_idea\_idx \longleftarrow$ random($nodes/4$) ;
    $idea\_idx \longleftarrow nodes * 3/4 +$ random($nodes/4$);
    $reward \longleftarrow 0$ ;
    $i \longleftarrow 0$;
    while $i < runs$ do
        $reward \longleftarrow reward+$ reward($node\_list[idea\_idx]$,
          $node\_list[anti\_idea\_idx]$, $graph$, $\lambda$, $\mu$, $messages$)
    return $\frac{reward*100}{runs*nodes}$
```

Algorithms 2 and 3 show how the initial seed nodes are calculated for the two cases we are interested in, namely one where the seed nodes have roughly equivalent numbers of connections and one where one seed node has a significantly higher number of connections than the other. We sort the nodes in the graph by their number of connections and either select one node at random and then a second node within 10% of the list of the first (Algorithm 2), or by selecting a node from the first quarter of the list and a second node from the last quarter of the list (Algorithm 3). Once the seed nodes are chosen, these algorithms perform a set number of runs through the system and then return the average reward over all the runs as a percentage.

Altogether this simulation tool can simulate a sub-class of social network infection models, which have only a single seed node for an idea and anti-idea.

5.1 Graph generation

We used the implementation of the generation algorithm of (1) (the Barabási–Albert model) in the networkX Python package[2] to generate our graphs. This algorithm starts out with m disconnected nodes forming a graph. It then adds nodes iteratively up to n nodes. Each new node is connected to m (in our case 3) random pre-existing nodes, which are selected preferentially (in the networkX package this means they are selected from a list of nodes where each node is represented by $k+1$ entries in the list where k is the number of existing connections the node already has). Barabási–Albert models are scale-free networks that share many topological properties with social networks.

5.2 Results

For each of our experiments we generated 100 random graphs with n nodes (for values of $n \in \{10, 100, 1000, 10,000\}$). We then ran Algorithms 2 and 3 one hundred times for runs of length n to $6n$ (i.e. the number of messages broadcast varied between the number of nodes in the graph up to

[2] https://networkx.org

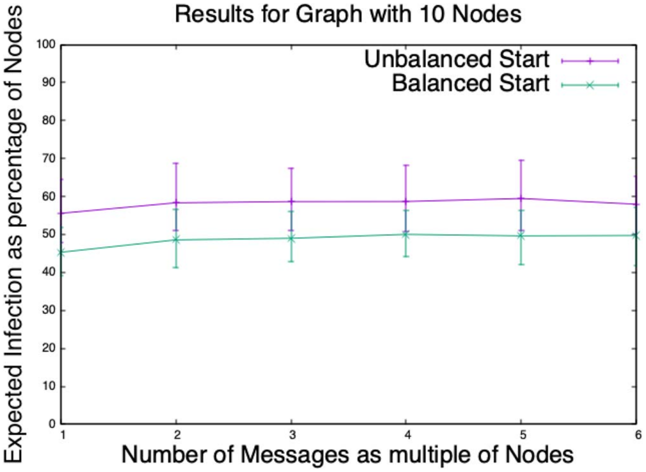

FIGURE 4. Graphs with 10 Nodes

six times the number of nodes in the graphs) for graphs with up to 1000 nodes and from n to $3n$ for the graphs with 10,000 nodes (for time reasons[3]) and then for each run length plotted the average reward for that run length.

The results are shown in Figures 4–6. The error bars indicate the 10th and 90th centile of the results across the 100 graphs. As can be seen the initial advantage of high connectivity (a 'gets there first' effect) continues to hold as the graph sizes increase. The implication of this is that our hypothesis that in large enough networks an initial advantage at a single seed node would not determine the speed at which an idea spread but that this would be depend instead upon the overall network structure (and so in experiments generating random network structures and seeding at random nodes within those we would see no particular advantage in expected reward) was incorrect. At least in Barabási–Albert networks, an initial early advantage (a single 'influencer' node if you will) is sufficient to ensure network dominance by an idea.

We also investigated the effect of varying how resistant agents were to changing their minds. We hypothesised that low values of μ would amplify the 'gets there first effect' since agents would be less likely to change their minds, away from the first idea they encountered. We investigated this for graphs of size 1000, where 3000 messages were exchanged in each run and λ set to 0.5 while μ (the probability of an agent changing their mind once infected by either an idea or its anti-idea) varied from 0.1 to 0.5. The results are shown in Figure 8. We were surprised to see that the value of μ apparently had little effect on the expected reward. This suggests that if sufficient messages are exchanged, the probability of an agent changing its mind on receipt of any individual has relatively little on the spread of an idea through the network—eventually, an agent will have received enough messages that it will have adopted the idea.

[3] It takes approximately $n*0.00001$ seconds to generate a graph of size n and approximately $n*0.0003$ seconds to generate one run of n messages on that graph.

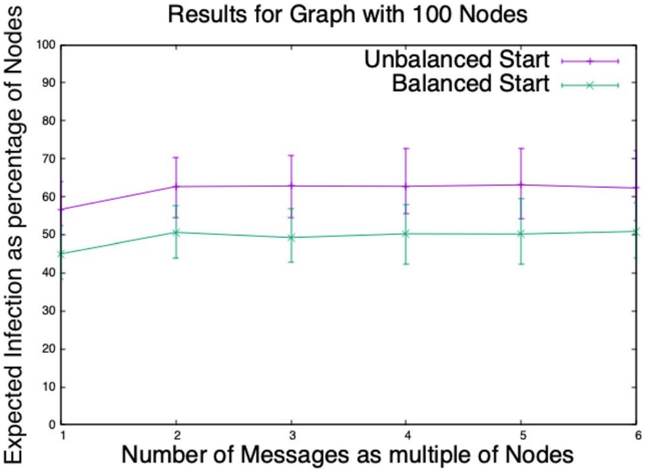

FIGURE 5. Graphs with 100 Nodes

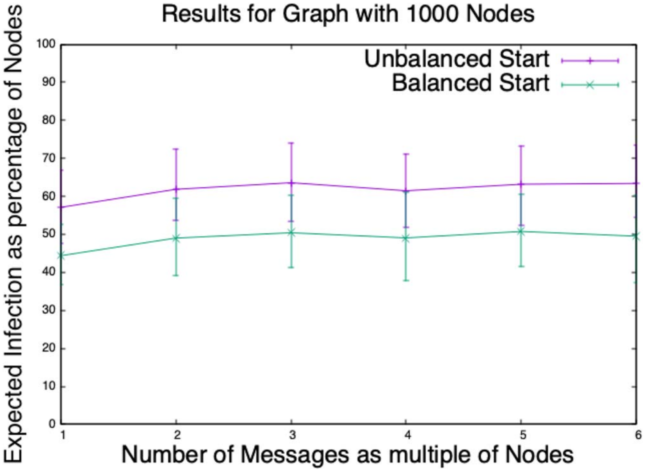

FIGURE 6. Graphs with 1000 nodes

6 Related Work

The influence of network structure on diffusion has been extensively studied in economics; see e.g. (21) for an extensive literature list and (16) for a more general overview of the impact of social network structure on behaviour.

The methodology used to study network structure impact on diffusion throughout the literature is numerical analysis, simulation and experiments. Both micro and macro aspects of the network structure have been considered, but in both cases these aspects refer to statistical properties of the network. For example, a macro network aspect example is the degree distribution in the network, while a micro network aspect example is the average distance between two agents in the network and network component diameters. In nearly all diffusion models, the likelihood of adopting new

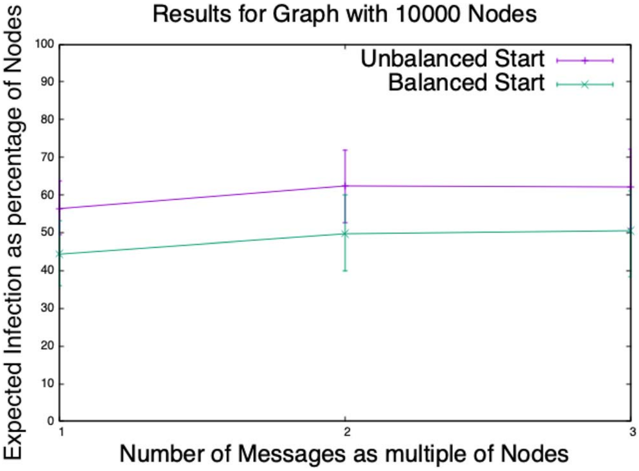

FIGURE 7. Graphs with 10,000 nodes

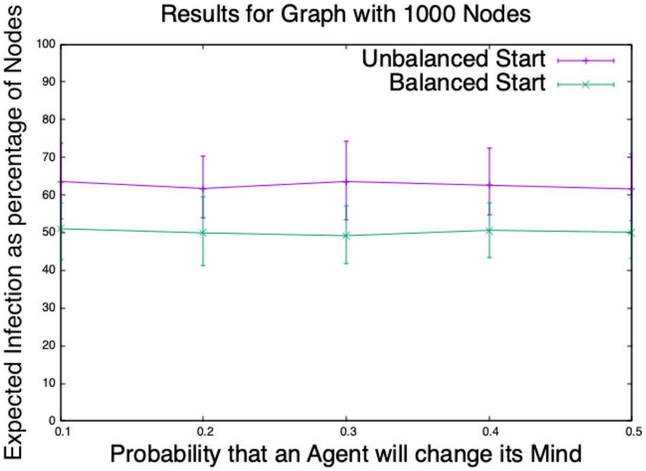

FIGURE 8. Graphs with 1000 nodes, varying μ

information or behaviour increases with the increase of adjacent agents who have adopted it and a higher agent degree leads to higher contagion (21). We also observe this here.

The approach in (4) is most similar to our own, using Markov chain models to capture the network structure and to show how opinions among the agents in the network may vary among a fixed set of opinions (a generalisation of the approach that assumes there is an idea, an ‘anti-idea’, and indifference that we use here). However, in their model the chance an agent will change its opinion does not depend upon its existing opinion, only upon the opinions of its neighbours. They use both formal analysis to generate results about the behaviour of the general system and Monte Carlo simulation to analyse a specific system consisting of a fully connected network and two possible opinions.

Model-checking information in social networks has been studied from a theoretical perspective in (25) and (9). Pardo and Schneider (25) consider the problem of verifying knowledge properties over social network models (SNMs) and show that the model-checking problem for epistemic properties over SNMs is decidable.

Dennis et al. (9) introduce a formal specification for SNM's and privacy properties that can be established to hold using model-checking using PRISM. Belardinelli and Grossi (3) present a model-checking algorithm and property specification logic for studying contagion-type models in open dynamic networks. This takes an agent view but does not explicitly consider the informational states of the agents. The proposed model-checking algorithm has not been implemented.

Kouvaros and LomuscioLomuscio (17) use parameterised model-checking in the MCMAS system to study opinion formation protocols for swarm robotics. These protocols are similar to threshold models and involve agents in a swarm switching their opinion to the majority opinion of their neighbours. The interest in this work was primarily on answering whether the protocol guaranteed convergence to an opinion, not on analysing the behaviour of information diffusion itself and probabilistic aspects were not studied.

Lastly, (32) use PRISM to evaluate the efficacy of methods for controlling harmful network propagation using different protection strategies for individual nodes. Although (32) are interested in security an protecting networks from e-viruses, the approach and methodology can be seen as related to ours in the case of information diffusion.

7 Discussion

We have developed a Markov chain-based framework for modelling information diffusion in social networks which takes an agent-centred view that includes an account of the agent's informational state when considering changes in the network. This framework represents a natural formalism for a variety of analysis techniques including model checking with the PRISM probabilistic model checker and Monte Carlo simulation.

Unfortunately, even comparatively simple models proved intractable for analysing models of interesting size in PRISM. However, Monte Carlo simulation allowed us to validate PRISM results for small networks on larger networks.

This suggests that results from simulation and model checking can be combined in a corroborative fashion, where model-checking results can be used to produce certain answers in small networks, backed up by statistical modelling corroborating those answers over larger networks, and confirms that Markov chains are a useful modelling tool in this area.

There are a number of interesting avenues for future work, particularly since the formalism presented here is comparatively simple. We would like to explore the use of Monte Carlo simulation in conjunction with the threshold models developed in (8). We would also be interested to investigate in more detail the choice of seed node, for instance using concepts of centrality, or looking at the effect of multiple seed nodes. Other avenues for investigation might be allowing the graph structure to change over time (e.g. as agents form connections with like-minded agents and abandon connections with agents they disagree with), looking at other techniques for generating social network graphs (e.g. those in (30)), or allowing a variety of agents with different reactions to challenging information (modelled by varying the values of μ and λ across the agents in our formalism)[4] . Ultimately, we would be interested in exploring whether network structure or actions

[4] Our thanks to an anonymous referee for many of these suggestion.

from network providers can influence information diffusion across networks by, for instance, preventing the formation of filter bubbles.

Open Data

The PRISM models, network graphs, output and timing data reported in this paper can all be found in the University of Liverpool Data Catalogue DOI: https://doi.org/10.17638/datacat.liverpool.ac.uk/909.

Simulator code and results can be found at https://github.com/louiseadennis/MonteCarloInformationDiffusion.git.

Acknowledgements

The research was partly supported by the project 'Better Video Workflows via Real-Time Collaboration and AI-Techniques in TV and New Media', funded by the Research Council of Norway (under grant no. 269790).

References

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, **74**, 47–97, 2002.

[2] N. Bailey. *The Mathematical Theory of Infectious Diseases and Its Applications*. Charles Griffin and Company Ltd., London, 1975.

[3] F. Belardinelli and D. Grossi. On the formal verification of diffusion phenomena in open dynamic agent networks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 237–245, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[4] P. Bolzern, P. Colaneri and G. de Nicolao. Opinion influence and evolution in social networks: a Markovian agents model. *Automatica*, **100**, 219–230, 2019.

[5] Z. Christoff and J. U. Hansen. A two-tiered formalization of social influence. In *Logic, Rationality, and Interaction*, D. Grossi, O. Roy and H. Huang., eds, pp. 68–81. Springer, Berlin Heidelberg, 2013.

[6] Z. Christoff, J. U. Hansen and C. Proietti. Reflecting on social influence in networks. *Journal of Logic, Language and Information*, **25**, 299–333, 2016.

[7] E. M. Clarke, O. Grumberg and D. Peled. *Model Checking*. MIT Press, 1999.

[8] L. A. Dennis and M. Slavkovik. Model-checking information diffusion in social networks with prism. In *Multi-Agent Systems and Agreement Technologies*, N. Bassiliades, G. Chalkiadakis and D. de Jonge., eds, pp. 475–492. Springer International Publishing, Cham, 2020.

[9] L. A. Dennis, M. Slavkovik and M. Fisher. "How Did They Know?" - Model-checking for analysis of information leakage in social networks. In *COIN-XII 2016 International Workshops, COIN@AAMAS, Singapore, May 9, 2016, COIN@ECAI, The Hague, The Netherlands, August 30, 2016. Revised Selected Papers*, pp. 42–59. Springer, 2016.

[10] M. Duflot, M. Kwiatkowska, G. Norman, D. Parker, S. Peyronnet, C. Picaronny and J. Sproston. Practical applications of probabilistic model checking to communication protocols. In *FMICS Handbook on Industrial Critical Systems*, S. Gnesi and T. Margaria., eds, pp. 133–150. IEEE Computer Society Press, 2010. To appear.

[11] M. Fisher, L. A. Dennis and M. Webster. *Verifying Autonomous Systems*. ACM Communications, 2013. Accepted for publication.

[12] V. Forejt, M. Kwiatkowska, G. Norman and D. Parker. Automated verification techniques for probabilistic systems. In *Formal Methods for Eternal Networked Software Systems (SFM'11)*, M. Bernardo and V. Issarny., eds, vol. **6659** of LNCS, pp. 53–113. Springer, 2011.

[13] U. Grandi, E. Lorini, A. Novaro and L. Perrussel. Strategic disclosure of opinions on a social network. In *Proceedings of the 16th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2017), International Foundation for Autonomous Agents and Multiagent Systems.*, 2017.

[14] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, **83**, 1420–1443, 1978.

[15] M. O. Jackson. *Social and Economic Networks*. Princeton University Press, Princeton, NJ, USA, 2008.

[16] M. O. Jackson, B. W. Rogers and Y. Zenou. The economic consequences of social-network structure. *Journal of Economic Literature*, **55**, 49–95, 2017.

[17] P. Kouvaros and A. Lomuscio. Formal verification of opinion formation in swarms. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '16*, pp. 1200–1208, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[18] M. Kwiatkowska, G. Norman and D. Parker. Stochastic model checking. In *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)*, M. Bernardo and J. Hillston., eds, vol. **4486** of LNCS (Tutorial Volume), pp. 220–270. Springer, 2007.

[19] M. Kwiatkowska, G. Norman and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV'11)*, G. Gopalakrishnan and S. Qadeer., eds, vol. **6806** of LNCS, pp. 585–591. Springer, 2011.

[20] M. Kwiatkowska and C. Thachuk. Probabilistic model checking for biology. In *Software Safety and Security*, NATO Science for Peace and Security Series - D: Information and Communication Security. IOS Press, 2014.

[21] P. J. Lamberson. Linking network structure and diffusion through stochastic dominance. In *Complex Adaptive Systems and the Threshold Effect, Papers from the 2009*. AAAI Fall Symposium, Arlington, Virginia, USA, November 5–7. AAAI Press, 2009.

[22] F. Liu, J. Seligman and P. Girard. Logical dynamics of belief change in the community. *Synthese*, **191**, 2403–2431, 2014.

[23] H. Mercier and D. Sperber. *The Enigma of Reason*. Harvard University Press, London, UK, 2017.

[24] M. E. J. Newman, D. J. Watts and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, **99**, 2566–2572, 2002.

[25] R. Pardo and G. Schneider. Model checking social network models. In *Proceedings of the 8th International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2017, Roma, Italy, 20–22 September 2017*, pp. 238–252. Open Publishing Assocation, 2017.

[26] T. Pedersen and M. Slavkovik. Formal models of conflicting social influence. In *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference, Lecture Notes in Computer Science, Nice, France, October 30 - November 3, 2017*, pp. 349–365. Springer, 2017.

[27] A. Rapoport. Spread of information through a population with socio-structural bias: I. assumption of transitivity. *The Bulletin of Mathematical Biophysics*, **15**, 523–533, 1953.

[28] T. Schelling. Dynamic models of segregation. *Journal of Mathematical Sociology*, **1**, 143–186, 1971.

[29] S. Smets and F. Velázquez-Quesada. How to make friends: a logical approach to social group creation. In *Logic, Rationality, and Interaction*, A. Baltag, J. Seligman and T. Yamada., eds, 2017. Forthcoming.

[30] R. Toivonen, J.-P. Onnela, J. Saramäki, J. Hyvönen and K. Kaski. A model for social networks. *Physica A: Statistical Mechanics and its Applications*, **371**, 851–860, 2006.

[31] C. Winship. Threshold models of social influence. In *The Oxford Handbook of Analytical Sociology*, P. Bearman and P. Hedström., eds. Oxford University Press, 2011.

[32] G. Zonghao, W. Ou, Y. Peng, H. Lansheng and W. Weiming. Model checking probabilistic network propagation protection strategies. In *The 2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 354–361. IEEE, 2016.

Received 10 February 2021