MODEL DESCRIPTION

As an experimental ground, the model is expected to provide two basic components. First, the model should be able to construct a social network that already has some measurable level of homophily present in its structure. Second, the model should be able to simulate a diffusion process, which is initiated from a small number of early adopters, over this social network. In this regard, the model is described in two parts: homophilious network evolution and diffusion.

Homophilious Network Evolution

The evolution process of the homophilious network is inspired by the famous "Segregation Model" (Schelling 1971). The process starts with a *small-world* network (Watts 1999; Wilensky 2005). In our network, each agent has a threshold value for her "happiness level". This level is related to the fraction of the friends with the same status in the agent's network neighborhood, as in Schelling's model. "Unhappy" agents (i.e. agents with fewer neighbors of their own status than what is desired according to the agent's happiness threshold) form ties with agents alike, and break ties with others.

%-similar-wanted is the model parameter that represents this personal threshold. After setting a value for *%-similar-wanted*, the routine given in Text Box 1 is activated for network evolution:

> *<u>Homophilious network evolution routine:</u>*
>
> *Create a "small world" network*
>
> *WHILE average "happiness level" is less than %-similar-wanted*
>
> *{Ask each agent:*
>
> *IF her "happiness level" is less than %-similar-wanted*
>
> *THEN form a new tie with a similar person from your friends' friendship networks*
>
> *AND dissolve a tie with a dissimilar friend of yours*
>
> *ELSE do nothing*
>
> *Calculate average "happiness level"}*
>
> *IF the network is connected THEN do nothing ELSE return to the beginning and create a new network*

In the network evolution routine, "happiness level" refers to the fraction of the friends with same status in an agent's friendship network. Note that all agents have the same threshold value for "happiness level" and that is equal to *%-similar-wanted*. In simple terms, the routine described by the pseudo-code in Text Box 1 works as follows: We start with a "small world" network. In each iteration, we evaluate the average happiness level of the agents. If it is below the set *%-similar-wanted* level, we ask each unhappy agent to find a new same status friend from the agents' friends' networks (i.e. friends' friends). At the same time, we ask the agent to 'unfriend' one of her current friends who has a different status than the agent itself. This way, the number of friends of the agent stays the same, whereas her social network gets more homophilious. Once this procedure is completed for all unhappy agents, we check the average happiness level again and the process goes on until we reach an average happiness level that is above the set *%-similar-wanted*. To ensure having a connected network at the end of this homophilious evolution process, the evolution process is repeated until the final network is a connected one.

The above heuristic ensures that the average degree of the social network stays the same as it develops into a more homophilious configuration as whenever an agent finds a new friend, she unfriends one, too. Nevertheless, in the end, agents are still heterogeneous in terms of their happiness levels and number of friends (i.e. their degree as a network node). Furthermore, the heterogeneity among agents even go deeper, as the agents also differ in the fundamental characteristics of their friendship networks, such as clustering coefficient, average path length, betweenness-centrality, and closeness-centrality.

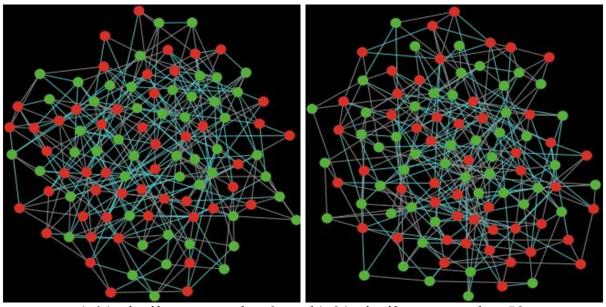
As the network evolves according to the set *%-similar-wanted* value, the homophily level of the network is measured with the average of the fractions of the same status friends in agents' friendship networks (*%-similar-total*).

Let *%-similar(i)* be the percentage of the friends with same status in agent i's personal network and *n* be the total number of people in the whole network. Then;

$$\%\_similar\_total = \frac{\sum_{i=1}^{n} \%\_similar(i)}{n}$$

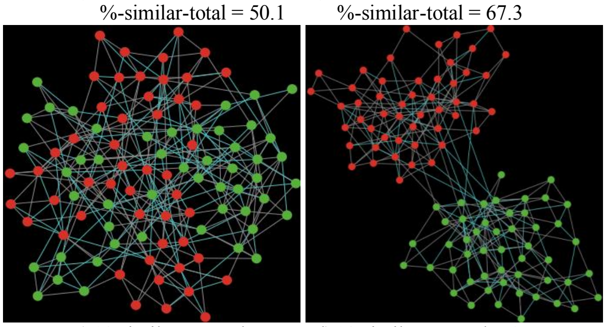
Although, there is no well-known and widely accepted measure for homophily to the best of our knowledge, there are examples of the one that is used here (Jackson 2008). Due to the initial random assignments of agents and their links *%-similar-total* is 50% on average in the beginning of the evolution process. This value changes as the network evolves homophiliously with respect to the value of *%-similar-wanted*.

When we let the social network in our model evolve according to the procedure described above, we obtain different levels of homophily in the network as a function of the *%-similar-wanted* value. As this parameter increases, the segregation patterns on the network become palpable. In *Figure 1*, we see patterns that are similar to the ones generated by Schelling's Segregation Model: as *%-similar-wanted* increases, network tends to become more segregated.



a) %-similar-wanted = 0
%-similar-total = 50.1

b) %-similar-wanted = 50
%-similar-total = 67.3



c) %-similar-wanted = 75
%-similar-total = 86.9

d) %-similar-wanted = 85
%-similar-total = 97.7

Diffusion

Homophily, besides its indirect role through shaping the network structure on which the diffusion takes place, also has a direct influence on the diffusion processes via information flows. As the focus of this study is to shed light on the indirect influence of homophily, we try to isolate it from the latter in our model. For that purpose, the network structure stays intact during the diffusion phase. In other words, we assume that the adopted behavior does not lead to homophilious tie formation or dissolution.

Diffusion starts with the initiation of the early adopters and takes off as people make decisions via social influence whether to adopt the innovation or not. The adoption mechanism in this study is an example of *threshold models* (Granovetter 1978). The most important factor in adoption decisions is the "adoption threshold". An agent adopts the innovation if the number of same status friends in the network neighborhood who adopted the innovation is more than the "adoption threshold". There are two ways of expressing the threshold, as an absolute number or as a fraction. Since our diffusion process includes making a decision about whether to adopt or not, expressing it as a fraction is preferred (Watts 2002). After the initiation step, in each round, the following heuristic begins to run until the adoption dynamics reach equilibrium:

Adoption routine:

Ask each agent:

IF (#-of-similar-friends-adopted / #-of-similar-friends) is greater than adoption threshold THEN

IF he has not adopted THEN make her adopt it ELSE do nothing

ELSE

IF he has not adopted THEN do nothing ELSE make her drop it

In simple terms, the routine described by the pseudo-code in Text Box 2 works as follows: Each time, we ask each agent whether the number of her same status friends who have already adopted the innovation is more then the adoption threshold, or not. If it is higher then the threshold, she adopts the innovation, or she remains an adopter if she has already adopted it. If the number of same status adopters is less than the threshold, then she does not adopt the innovation, or quits it if she is already an adopter.

Before running the simulation, we set the key parameters that define the network structure, such as *%-similar-wanted* level, average degree of the network, and the number of agents in the network. Afterwards, we first run the homophilious network evolution routine. During this time it is possible to see how the network evolves. At the end of this process, we record key network statistics and the overall homophily level. Then, we run the diffusion routine and study the diffusion dynamics.[i]

Notes

[i] During model development, the model is tested for both conceptual and technical problems. In the context of verification testing, detailed code walkthroughs and extreme condition simulations are performed both on the whole model and its sub-components in isolation. As a structural validation task, we tested the behavior of individual agents under various conditions. The test conditions are chosen as the ones that allow us to predict the proper behavior that the agents are expected to demonstrate. As a result of these inspections and tests, we concluded that the model performs well with respect to the conceptual boundaries and research questions considered in this study. Since the model is a generic model that is intended to serve as a dynamic exploration platform, which does not correspond, to a particular innovation or social network, it was not possible to compare the macro level model behavior with empirical data.