Overview, Design Concepts, and Details (ODD) for the Axelrod's model for Cultural Dissemination

Arezky Hernández Rodríguez
Academia de Matemáticas, Universidad Autónoma de la Ciudad de México (UACM), México DF. (arezky@gmail.com)

Abstract. Following is the ODD description of the Axelrod's model of Cultural Dissemination [1].

1. Purpose:

The Axelrod's model of cultural dissemination [1] is an agent-model designed to investigate the dissemination of culture among interacting agents on a society.

2. Entity, State Variables and Scales:

Axelrod model consists in a population of agents, each one occupying a single node of a square network of size L. There are $L \times L$ agents. The culture of an agent is described by a vector of F integer variables called *features*. Each feature consists in a nominal variable that can assume q different values (called *traits*) between 0 and $q - 1$. In the original Axelrod's model the interaction topology is regular bounded (non-toroidal), each agent can interact only with its four closest neighbors (von Neumann neighborhood) and agents can not move.

In the present implementation of the model using NetLogo [2], each agent is located at each patch of the grid with the default agent shape. We have also set not toroidal boundaries, but the simulation properly functions as well in the toroidal case (try it!). In order to be able to go beyong the original Axelrod model we have included a parameter *radius* which can take positive real values to explore the implications of other neighborhood sizes. The agent neighborhood set is obtained including all neighbors at distance less or equal than *radius* where *radius* is used as a distance from the agent own coordinates. Then, when the agent is exactly in the center of its patch, the original Axelrod's four von Neumann agent neighbors are obtained when using $radius = 1$ while the original Axelrod's eight Moore agent neighors are obtained when setting $radius = \sqrt{2}$. Extended neighbors set can be obtained for higher *radius* values (try it!).

It is also implemented the possiblity for agents to move according to three parameters: *veloc*, *steplength* and *angle*. If moving, select the velocity of agent movement with *veloc* greater than zero, select the length of the step with *steplength* and the angle of rotating with *angle*. At each time step agents decide to move taking *veloc* as a probability. In case of actual movement, agents select at random a value between zero and *angle*. If the value is in the upper half of *angle* it is added (as and angle) to the current direction of the agent. If it is in the lower half it is subtracted. Ones a direction is selected, agent moves a distance *steplength*.

For visualization, each agent adopts a color representing its culture. This is done taking the agent culture traits as a number, computing the corresponding number of base q and mapping this number to a color number in the blue range between 100 and 109.9.

3. Process Overview and Scheduling:

Time is modeled as discrete time steps and the updating of agent variables are asyncronous-random: at each iteration all agents are fired in a random order (newly randomised each time as is standard in NetLogo). Each fired agent i follows the following steps:

(i) agent decides to move according to the probability *veloc*. If moving
   (a) select a direction according to the value of *angle*.
   (b) select a step distance according to *steplength*.
   (c) agent moves

(ii) checks if it is active and in this case
   (a) selects a neighbor for cultural interaction (without any constraint)
   (b) interact with the selected neighbor with probability proportional to the amount of features they have with the same trait values, that is the amount of features they share.
   (c) if interaction, the active agent modifies one of its cultural features he does not share with selected neighbor copying the neighbor cultural trait value. Note that if there are no common traits value, the interaction is not possible and the respective agents refuse to influence each other.
   (d) agent i modifies it color according to its new cultural values.

It is possible to follow the evolution of the system according to the agent colors (related with the corresonding cultural values) and according to the monitors which report the Number of Cultures, the Number of Possible Interactions, the Number of Real Interactions and the Number of Active Agents. The time evolution of the system is reported on the Graph where normalized values are included.

The 'Go Forever' button stops when the Number of Active Agents reaches zero.

4. Design Concepts:

4.1. Basic Principles:

Axelrod's model for Cultural Dissemination have been studied extensively in scientific literature [3, 4, 5, 6, 7, 8, 9, 10, 11]. The model descibed an implemented here is a basic implementation from which other extensions have been developed. The dynamics are based on two main mechanisms:

(i) agents tend to chose culturally similar neighbors as interaction partners (*homophily*) and

(ii) during interaction agents influence each other in a way that they become more similar.

4.2. Emergence:

The interplay of these mechanisms either leads to cultural homogeneity (all agents are perfectly similar) or the development of culturally distinct regions (multicultural society). The model allows studying to which degree the likelihood of these two outcomes depends on the size of the population, the number of features the agents hold, the number of traits (values) each feature can adopt and the neighborhood size (interaction range).

4.3. Adaptation:

Neighbor agents interact with probability proportional to the amount of features they share with the same trait values and after an actual interaction, they share now one more feature. Then, the interaction probability between any two agents changes in time. Particularly, when any two agents are completely different the interaction is stopped and if they are completely equal there is nothing new to copy one from the other and the interaction is also stopped. The process outlined above continues until no cultural change can occur. The dynamics then reaches an absorbing state which is one of the two possible final states. This happens when every pair of neighboring agents have cultures that are either identical or completely different.

4.4. Objetive:

In this model there are no objectives for the agents as such. Agents do not act deliberately to obtain any benefit.

4.5. Learning:

The model in the present form does not have any learning component.

4.6. Prediction:

There is no prediction component in the present model.

4.7. Sensing:

When an agent becomes a focal agent, one of the focal agents neighbors is selected at random. Neighbor agents are those which are at a distance less or exactly equal to that of the value of the parameter *radius*. If $radius = 1$, then it is von Neumann neighborhood (closest agents at north, south, east and west).

4.8. Interaction:

Each fired agent i follows the following steps:

(i) Check if it is active:
counts the amount of neighbors which overlap fulfills $0 < O(i, j) < F$ where the overlap is the amount of features agent i and neighbor j have with the same trait value. If the the amount of neighbors is greater than zero, then agent i is an active agent and the active agent counter is increased by one. If there are no neighbours fullfilling the above condition, then agent i is not active.

(ii) If agent i is active, then it is selected a neighbor for cultural interaction. Note that it could be selected a neighbor with zero of full overlap ($O(i, j) = F$).

(iii) active agent i and neighbor j interact if $0 < O(i, j) < F$ and with probability proportional to the overlap. That is

$$P(i, j) = \frac{O(i, j)}{F} \tag{1}$$

If $0 < P(i, j) < 1$, then the counter for possible interaction is increased by one.

(iv) In case of actual interaction, the counter of real interaction is increased by one. The interaction consists of selecting at random one of the features on which the two agents differ and changing the active agents feature to the interaction partners trait.

The dynamics stops when the number of active agents reaches zero. This procedure is different to that of the original Axelrod's model where the dynamics stop when all pair of neighbor agents share all of none of its features. The concept of active agent was first intruduced in Ref. [12]. This procedure strongly increases the efficiency of the dynamical evolution of the system and allows to save computational time.

4.9. Stochasticity:

Stochasticity in the model is given in different forms:

(i) It is choosen uniformly at random from $\{0, 1, 2, ..., q - 1\}$ traits values for each agent at the beginning.

(ii) At each time step (tick) agents update its cultural value in an asyncronous-random updating. That is the computer makes a list where all agents are included in a random order and the list is followed until all agents are choosen (becoming a focal agent each time).

(iii) When agents interact, the focal agent select stochastically a neighbor.

4.10. Collectives:

When the dynamics stops there are no active agents. Using the button 'Report Final State' the program makes different visible networks which include all neighbors agents with the same culture.

4.11. Observation:

In the model interface there is a graph reporting the number of different cultures on the society, the number of possible and real interactions, and the number of active agents. All of them normilized values.

A possible interaction is that were both agents share more than zero and left than all its features. That is when the probability of interaction meets $0 < P(i,j) < 1$.

A real interaction is when focal agent actualy changes the value of one of its features.

An active agent is one which has at least one neighbor where the interaction is possible.

All this values are normilized to have values between zero and one: the number of cultures is divided by q^F, while the number of possible an real interactions are divided by the number of agents which is L^2.

There are also different monitors which report the same values using natural number for better following the system dynamics.

Simulation stops when the number of active agents reaches zero. That means that each agent share all or none of its traits value with each of its neighbors.

The user can follow the time evolution of the system using the 'Go Once' button of the 'Go Forever' one. The 'Go Forever' button can be pushed againt to stop the simulation at any time.

The user can also see the neighbor set of a randomly selected agent using the 'watch neighbors!' button. It is usefull to see the implications of different values of the parameter 'radius'. When pushing the button one agent is selected at random and marked as red color and its corresponding neighbors marked as green color. For correct visualization the button should be used when the dynamics is stopped, but not necessary at the begining or the end. Interesting results are obtained when agents are allowed to move. Try it!.

The button 'Report Networks' reports the number of cultural regions in the population and the number of agents in the biggest one (also normalized). A region is a set of neighbor agents that are similar on all features. That is that they share exactly the same culture. To obtain the original Axelrod's model results the 'Report Networks' should be used when the dynamics stops and there are no active agents.

When counting the number of cultural domains it is considered that two domains are different if they are not connected, even if agents in both domains share same culture.

Note also that two agents could have close (but with zero overlap) cultural values and then its corresponding colors could be so similar that it could induce to think that their cultures are the same. Just check to see that it is not.

The increment in the *veloc* parameter is very small (0.01) allowing fine tunning of the velocity. You can see odd results for very low velocities. Try it!.

4.12. Initialization:

Initially, agents traits are chosen uniformly at random from the set $\{0, 1, 2, ..., q - 1\}$. Also the directions of agents are randomly chosen.

4.13. Input Data:

There are not input data sets.

4.14. Submodels:

A submodel is used for calculating the number of cultural regions in the population and the number of agents in the biggest one. This submodel is implemented with the button 'Report Networks'. The procedures used are the following:

(i) Build networs: It is linked all agents of the same culture:
Each agent looks for a neighbor which is in its neighborhood (agent inside in radius) and if the overlap between then are F it is created an undirected links.

(ii) Find all the components of the networks:
It is implemented the submodel 'find-all-components' of the Model 'Giant Component', which can be found in the Netlogo Model Library, section Networks. The 'find-all-component' submodel explores all disconnected networks counting all its corresponding nodes. It is a recursive submodel.

References

[1] R. Axelrod, Journal of Conflict Resolution **41**, 203 (2005).
[2] U. Wilensky, *NetLogo. http://ccl.northwestern.edu/netlogo/* (Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, ADDRESS, 1999).
[3] A. Flache and M. W. Macy, Journal of Conflict Resolution **55**, 970 (2011).
[4] J. C. González-Avella *et al.*, Phys. Rev. E **73**, 046119 (2006).
[5] K. Klemm, V. M. Eguíluz, R. Toral, and M. S. Miguel, Physica A **327**, 1 (2003).
[6] K. Klemm, V. M. Eguíluz, R. Toral, and M. S. Miguel, Phys. Rev. E **67**, 045101(R) (2003).
[7] K. Klemm, V. M. Eguíluz, R. Toral, and M. S. Miguel, Phys. Rev. E **67**, 026120 (2003).
[8] J. C. González-Avella, M. G. Cosenza, and K. Tucci, Phys. Rev. E **72**, 065102(R) (2005).
[9] K. I. Mazzitello, J. Candia, and V. Dossetti, Int. J. Mod. Phys. C **18**, 1475 (2007).

[10] J. Candia and K. I. Mazzitello, J. Stat. Mech. P07007 (2008).
[11] J. C. González-Avella *et al.*, J. Artif. Soc. Soc. Simul. **10**, 9 (2007).
[12] L. A. Barbosa and J. F. Fontanari, Theor. Biosci. **128**, 205 (2009).