

Testing and Validating Models

EES 4760/5760

Agent-Based and Individual-Based Computational Modeling

Jonathan Gilligan

Class #7: Thursday, September 12 2019

Organization

Organization

- Download culture dissemination model from the Download page on the course web site, https://ees4760.jgilligan.org/files/models/chapter_06/CultureDissemination_Untested.nlogo, or Brightspace.
 - The paper describing the culture dissemination model is also on the Download page, https://ees4760.jgilligan.org/files/models/chapter_06/axelrod_culture_dissemination_1997.pdf, and Brightspace.
- Sit together in pairs, with a partner.
- Team Projects:
 - Class Presentations October 10
 - Choice of two projects:
 - Adaptive behavior (Business investor model, Ch. 10)
 - Agent interactions (Telemarketer model, Ch. 13)
 - Decide which one you want to do and choose a partner (undergrads with undergrads, grads with grads)

Finding and Fixing Errors

Classes of Errors

- Typographical (typing `pxcor` when you mean `pycor`)
- Misunderstanding NetLogo language:

```
ask turtle 5 [  
  let neighbor-patches patches in-radius 2  
  ask neighbor-patches [set pcolor green]  
]
```

versus

```
ask [patch-here] of turtle 5 [  
  let neighbor-patches patches in-radius 2  
  ask neighbor-patches [set pcolor green]  
]
```

- Wrong display settings (wrapping)
- Run-time errors (e.g., division by zero, forgetting to initialize globals, etc.)
- Logic errors (**hard to find**)
- Formulation errors (**hard to find**)

Independent Re-Implementation of Submodels

- If your model needs a tricky calculation:
 - Try it in another format: spreadsheet, scripting language (Python, R, Matlab, etc.)
 - Compare to NetLogo results

Culture-Diffusion Model

Culture-Diffusion Model

- **Entities and State Variables:**
 - Each patch is a village (10×10)
 - Culture is characterized by 5 numbers (integers 0–9)
 - example: 58354
 - Similarity of two villages = (# matching numbers) / 5
 - Goes from 0–1
 - 04976 and 44873 have a similarity of 0.4 (2 matches)

Culture-Diffusion Model

- **Process Overview**

1. *Cultural interaction*

- Each tick *one* random village is active.
- Active village picks a random partner from neighbors sharing an “edge.”
- Maybe the active village interacts with partner
 - Probability of interacting = similarity.
 - The more similarity, the more likely to interact
 - Two villages that have nothing in common won't interact.
- If they interact, active village copies one of the partner's culture numbers.

2. *Output*: Update patch colors and graphs

3. If no patches have interacted for 1000 ticks, it stops.

Expected behavior

- We expect clusters of villages with different cultures, where villages in a culture become increasingly similar.
- Color patches according to mean similarity with neighbors:
 - Compare to four neighbors: up, down, left, right
 - black if mean *similarity* = 0 (nothing in common with any neighbor),
 - white if *similarity* = 1 (identical to all neighbors),
 - shades of red in between.

New NetLogo Primitives

- `myself`:
 - `self` refers to the current turtle or patch
 - `myself` refers to the turtle or patch that asked the current turtle or patch to do something.

```
ask turtles-here [ set color [color] of myself ]
```

- `neighbors4` of a patch gives the four neighboring patches that share an edge (up, down, right, left, but not diagonal).
 - A patch on an edge or corner will have fewer than four neighbors.

Test the model

Work with your partner to examine the culture diffusion model.

- Run it and see what happens.
 - Look for weird behavior that might indicate an error.
 - Save data to files with “test-output-on?” switch.
- Examine the model code
- Inspect agents. Use the `show-similarities` procedure in the agent monitor.
- Announce to the class if you find an error