LABs Presentation & Introduction to Agent Based Modeling with Python MESA

Lorenzo Ghiro

lorenzo.ghiro@unitn.it

LABs Goal

- Provide experience in the implementation of typical algorithms used in distributed systems.
- Gain experience with standard tools and frameworks for
  - distributed programming
  - modeling & analysis of distributed systems
- Hands-on daily used, popular, really useful libraries
- Have fun! ...that's why Python as main programming language :)

I❤PYTHON

Schedule

UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department

Date	Content	Notes
24th Sep	Labs presentation, Agent Based Modeling -> intro to MESA	https://mesa.readthedocs.io/en/master
1st Oct	More tutorials/examples on MESA	
8th	In class exercise, assignment!	
15th		
22th	Complex networks analysis with NetworkX	https://networkx.org
29th	More tutorials on NetworkX	
5th Nov		
12th	Distributed Programming with RAY	https://ray.io
19th	More tutorials on Ray	
26th		
3rd Dec	Build a blockchain in Python!	
10th		

Good to know

- The final grade will be given 50% by the lab projects and 50% by a final (oral) examination.
- Labs website:

  https://lorebz.github.io/labsdistributedsystems2
- Course website:

  http://cricca.disi.unitn.it/montresor/teaching/ds2
- Meeting me... send me an email :)

  I do not live in Trento but can arrange a meeting somehow if necessary :)

My Python setup and tools

- I work on Ubuntu 20.04
- Python 3.7
  - Anaconda + pip
- PyCharm + SublimeText3
- Sometimes Jupyter Notebooks/Lab

You are free to use any other editor and work on other OSs... this sidenote is just to say that provided code has been tested only under this setup

Maybe useful installation notes

- Install Anaconda

  https://docs.anaconda.com/anaconda/install/linux

- Anaconda with Python 3.7

  https://www.anaconda.com/blog/python-3-7-package-build-out-miniconda-release

- `sudo snap install pycharm-community --classic`

Agent Based Modeling

- Agent-based modeling (ABM) is a way to simulate the behaviors and interactions of autonomous entities over time.
- Agents:
  - have properties and behaviors.
  - interacts with and influence each other.
  - learn from their experiences.
  - adapt their behaviors to they are better suited to their environment(s).
- Example: SIR models perfect for ABM.

ABM tools

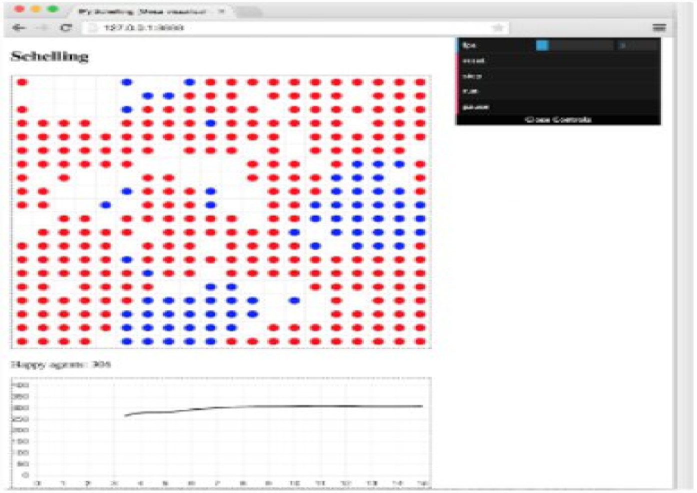
- Agent-based modeling has been used successfully to model complex adaptive systems.
- Biology, Supply chains, economics, military planning, consumer market analysis, **Distributed Systems/Algorithms**!
- ABM tools
  - StarLogo, NetLogo, Swarm, MASON, EcoLab, GAMA, Repast...
  - **MESA**

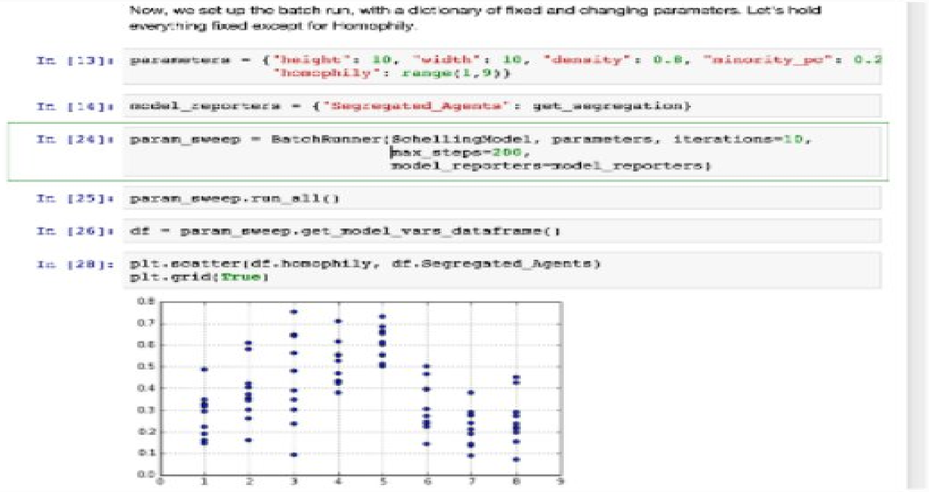
Kazil, Jackie, David Masad, and Andrew Crooks. "Utilizing **Python** for **Agent-Based Modeling**: The **Mesa** Framework." *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, Cham, 2020.

Mesa: Agent-based modeling in Python

Mesa is an Apache2 licensed agent-based modeling (or ABM) framework in Python.

It allows users to quickly create agent-based models using built-in core components (such as spatial grids and agent schedulers) or customized implementations; visualize them using a browser-based interface; and analyze their results using Python's data analysis tools. Its goal is to be the Python 3-based counterpart to NetLogo, Repast, or MASON.





Now, we set up the batch run, with a dictionary of fixed and changing parameters. Let's hold everything fixed except for Homophily.

```
In [13]: parameters = {"height": 10, "width": 10, "density": 0.8, "minority_pc": 0.2
                       "homophily": range(1,9)}

In [14]: model_reporters = {"Segregated_Agents": get_segregation}

In [24]: param_sweep = BatchRunner(SchellingModel, parameters, iterations=10,
                                   max_steps=200,
                                   model_reporters=model_reporters)

In [25]: param_sweep.run_all()

In [26]: df = param_sweep.get_model_vars_dataframe()

In [28]: plt.scatter(df.homophily, df.Segregated_Agents)
         plt.grid(True)
```

Above: A Mesa implementation of the Schelling segregation model, being visualized in a browser window and analyzed in an IPython notebook.

Mesa: Agent-based modeling in Python

Getting started quickly

```
pip install mesa
```

clone the repository folder; invoke `mesa runserver` for one of the examples/ subdirectories

```
mesa runserver examples/wolf_sheep
```

- Following tutorial together in class
  - Mesa Introductory Tutorial
  - Mesa Advanced Tutorial

Questions?