ABM Frameworks

Session IV - Introduction to ABM

Ali Seyhun Saral (Uni. Bologna)

16 June 2022

Agent-Based Model Frameworks

They can help you:

- Create a world (grid, network, etc.)
- Shorten the development time
- Run faster simulations

Frameworks

List of frameworks-Wikipedia

Python Frameworks

- Repast4py
- Mesa
- AgentPy

AgentPy framework

- Simple Syntax
- Beginner Friendly
- Jupyter Friendly

AgentPy Components

- `Agent`
- Environments
  - `Grid`: A matrix of cells
  - `Network`: A network of nodes
  - `Space`: A continous space
- `Model`
- `Experiment`

Creating an Agent

```python
import agentpy as ap

class MyAgent(ap.Agent):
    def setup(self):

        self.payoff = 0

    def do_something(self):
        self.payoff += 1
```

- `.setup()` is like `__init__` but it automatically inherits from `ap.Agent` variables. (i.e. no need to call `super().__init__()`)
- Just like `__init__()`, it will be called automatically when the agent is created.

Automated links to simulation objects

- `myagent.model`: Model instance agent is in
  - For instance `myagent.model.grid`: Grid instance agent is in
- `myagent.id`: Agent ID
- `p`: Parameter dictionary
- `log`: Recorded variables

Model

```python
class MyModel(ap.Model):

    def setup(self):
        """ Initiate a list of new agents. """
        self.agents = ap.AgentList(self, self.p.num_agents, MyAgent)

    def step(self):
        """ Call a method for every agent. """
        # do something during the round
        self.agents.do_something()

    def update(self):
        """ Record a dynamic variable. """
        # do something at the end of the round
        self.record('average_payoff', sum(self.agents.payoff)/len(self.agents))

    def end(self):
        """ Report an evaluation measure. """
        self.report('my_measure', 1)
```

```python
parameters = {'num_agents': 10, 'steps': 4}
new_model = MyModel(parameters)
results = new_model.run()
```

```
Completed: 1 stepsCompleted: 2 stepsCompleted: 3 stepsCompleted: 4 steps
Run time: 0:00:00.002193
Simulation finished
```

Results

```python
print(results)
```

```
DataDict {
'info': Dictionary with 9 keys
'parameters':
    'constants': Dictionary with 2 keys
```

```
'variables':
    'MyModel': DataFrame with 1 variable and 5 rows
'reporters': DataFrame with 2 variables and 1 row
}
```

. . .

```
print(results.info)
```

```
{'model_type': 'MyModel', 'time_stamp': '2022-06-19 11:27:54', 'agentpy_version': '0.1.5', 'p
```

. . .

```
print(results.variables.MyModel)
```

```
   average_payoff
t
0             0.0
1             1.0
2             2.0
3             3.0
4             4.0
```

Model components: `setup(self)`

```
class MyModel(ap.Model):

  def setup(self):
    self.agents = ap.AgentList(self, self.p.num_agents, MyAgent)
    print("hi")
    print(self.agents)
```

- Create the agents

`ap.AgentList(self, num_agents, agent_class)`

- We no longer need to create a loop for agent creation. `AgentList` does it by itself.

Model components: `step(self)`

```
class MyModel(ap.Model):
# ...
    def step(self):
        """ Call a method for every agent. """
        # do something during the round
        self.agents.do_something()
# ...
```

- This is where we do the main work of the model. What happens in each round?
- We no longer have to run a loop over all the agents.
- `self.agents.do_something()` runs `do_something()` for every agent.

Model components: `update(self)`

```
class MyModel(ap.Model):
# ...
def update(self):
        """ Record a dynamic variable. """
        # do something at the end of the round
        self.record('average_payoff', sum(self.agents.payoff)/len(self.agents))
# ...
```

- This is where we record the dynamic variables.
- If we need something to calculate only at the end of the round, this would be the place.

Model components: `end(self)`

```
class MyModel(ap.Model):
#...
    def end(self):
        """ Report an evaluation measure. """
        some_measure = calculate_measure()
        self.report('my_measure', some_measure)
```

- You can run things at the end of the simulation

- You can record measures belong to the general simulation at the end. Such as summary statistics.

Running a simulation

- Model parameters are defined in a dictionary.
- Then it is plugged into a model

```
parameters = {
    'my_parameter':42,
    'agents':10,
    'steps':10
}

model = MyModel(parameters)
```

. . .

- Then the model is run by model.run(). It returns a `Results` object which contains useful information about the simulation.

```
results = model.run()
```

Model Procedure

1. The model initializes with the time-step `Model.t = 0`.
2. `Model.setup()` and Model.update() are called.
3. The model's time-step is increased by 1.
4. `Model.step()` and `Model.update()` are called.
5. Step 2 and 3 are repeated until the simulation is stopped.
6. `Model.end()` is called.

https://agentpy.readthedocs.io/en/latest/overview.html

Stopping the model

- Calling the `Model.stop()` during the simulation.
- Reaching the time-limit, which be defined as follows:
  - Defining steps in the paramater dictionary.
  - Passing steps as an argument to `Model.run()`.

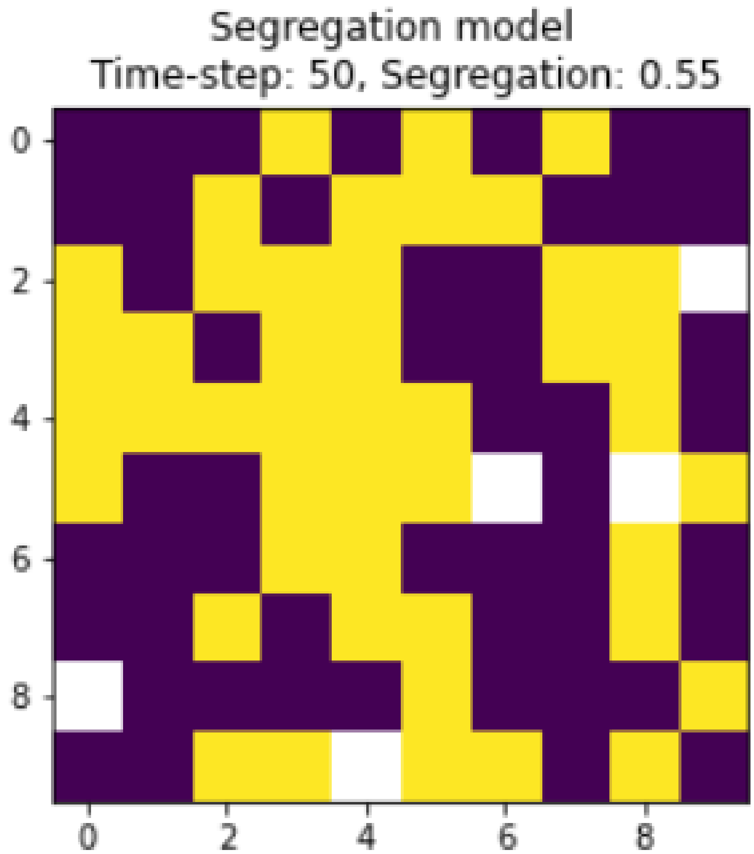
```
parameters = {
    'my_parameter': 42,
    'agents': ap.IntRange(10, 20),
    'steps': ap.IntRange(10, 20)
}
sample = ap.Sample(parameters,n=10)
print(sample)
```

```
Sample of 100 parameter combinations
```

Practice

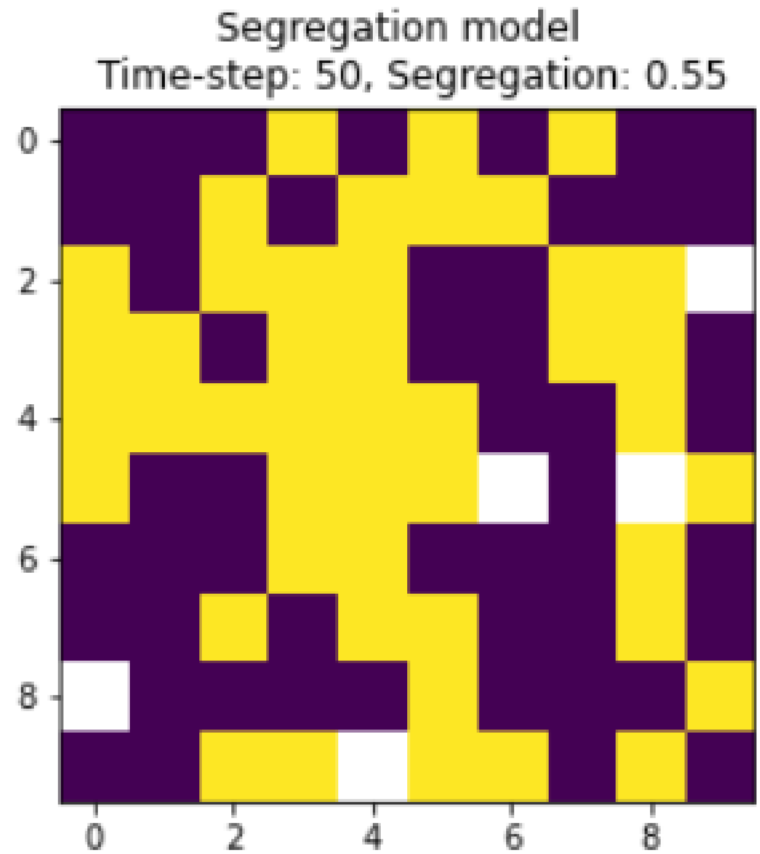
`Ex_agentpy1.ipynb`

Spatial ABM's



- Spatial ABM's are models that use a space to represent the world.
- Usually a torus/donut shape is used.
- Can be:
  - Continuous
  - Gred
  - Network

Schelling's Seggregation Model



-

- 20x20 grid
- Each cell is a house in the grid
- Each agent live in a grid
- There are two races
- Each agent is happy at least a proportion of agents are like them

Practice

`Ex_agentpy2.ipynb`

Consistency Analysis

- How many times should I repeat the model?

(Lee et al. 2015) Coefficient variation: ratio of SD. of a sample to its mean.

$$c_v = \sigma/\mu$$

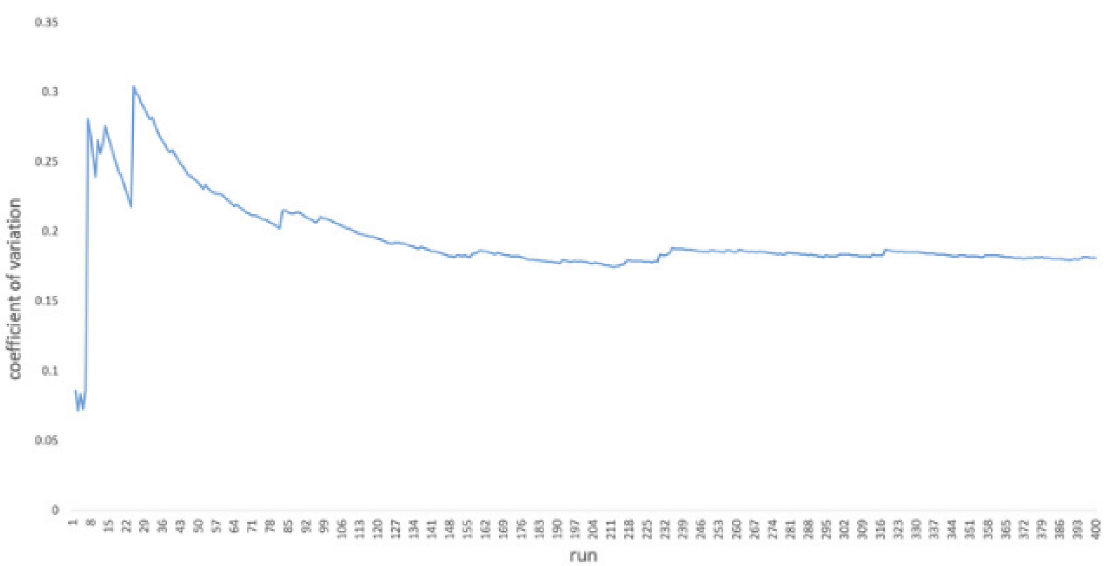
. . .

For increasing sample size calcuate the difference:

$$E = c_v^501 - c_v^500$$

Set a threshold of E. (0.01 is common)

Consistency Analysis



From: Introduction to Agent-Based Modeling, By Marco A. Janssen

Sensitivity analysis

- How robust are model results to changes parameters?
- Sensitivity analyses are techniques for estimating the effect of parameters on the simulation.

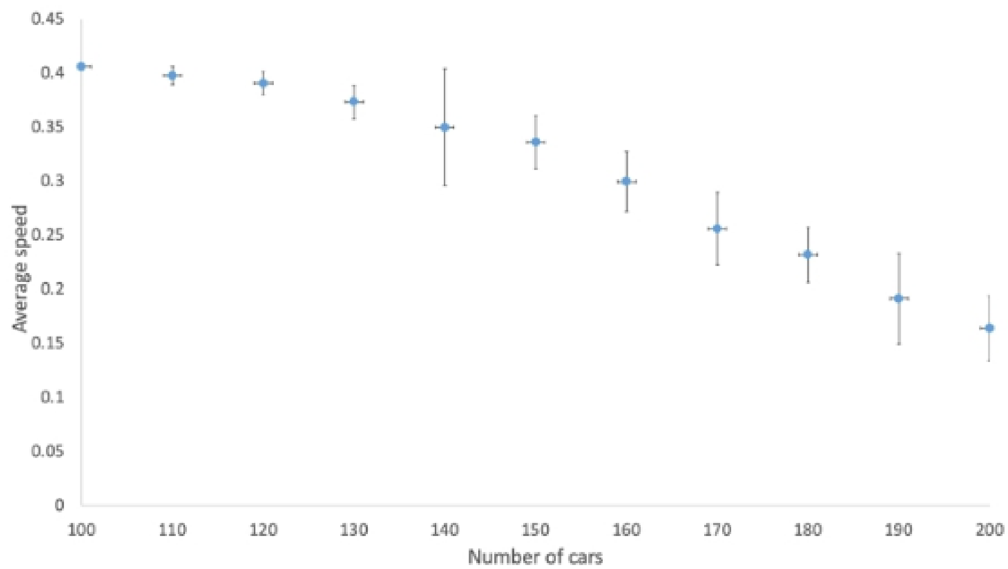
. . .

- Local sensitivity analysis : (Cariboni et al. 2007)
  - The effect of a single parameter on the simulation.
  - Vary one parameter at a time.

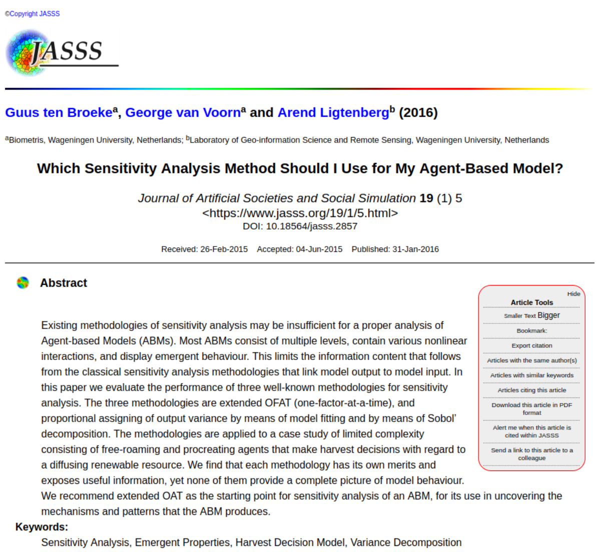
. . .

- Global Sensitivity Analysis
  - The effect of multiple parameters on the simulation.
  - Sample from simulations with different values of parameters.
  - For instances, regression based sensitivity analysis (Downing, Gardner, and Hoffman 1985)

Sensitivity analysis



Reading List


©Copyright JASSS

JASSS

Guus ten Broeke[a], George van Voorn[a] and Arend Ligtenberg[b] (2016)

[a]Biometris, Wageningen University, Netherlands; [b]Laboratory of Geo-information Science and Remote Sensing, Wageningen University, Netherlands

Which Sensitivity Analysis Method Should I Use for My Agent-Based Model?

Journal of Artificial Societies and Social Simulation **19** (1) 5
<https://www.jasss.org/19/1/5.html>
DOI: 10.18564/jasss.2857

Received: 26-Feb-2015 Accepted: 04-Jun-2015 Published: 31-Jan-2016

Abstract

Existing methodologies of sensitivity analysis may be insufficient for a proper analysis of Agent-based Models (ABMs). Most ABMs consist of multiple levels, contain various nonlinear interactions, and display emergent behaviour. This limits the information content that follows from the classical sensitivity analysis methodologies that link model output to model input. In this paper we evaluate the performance of three well-known methodologies for sensitivity analysis. The three methodologies are extended OFAT (one-factor-at-a-time), and proportional assigning of output variance by means of model fitting and by means of Sobol' decomposition. The methodologies are applied to a case study of limited complexity consisting of free-roaming and procreating agents that make harvest decisions with regard to a diffusing renewable resource. We find that each methodology has its own merits and exposes useful information, yet none of them provide a complete picture of model behaviour. We recommend extended OAT as the starting point for sensitivity analysis of an ABM, for its use in uncovering the mechanisms and patterns that the ABM produces.

Hide
Article Tools
Smaller Text Bigger
Bookmark:
Export citation
Articles with the same author(s)
Articles with similar keywords
Articles citing this article
Download this article in PDF format
Alert me when this article is cited within JASSS
Send a link to this article to a colleague

Keywords:
Sensitivity Analysis, Emergent Properties, Harvest Decision Model, Variance Decomposition


JASSS

Ju-Sung Lee[a], Tatiana Filatova[a], Arika Ligmann-Zielinska[b], Behrooz Hassani-Mahmooei[c], Forrest Stonedahl[d], Iris Lorscheid[e], Alexey Voinov[a], Gary Polhill[f], Zhanli Sun[g] and Dawn C. Parker[h] (2015)

[a]University of Twente, The Netherlands; [b]Michigan State University, United States; [c]Monash University, Australia; [d]Augustana College, United States; [e]Hamburg University of Technology, Germany; [f]The James Hutton Institute, United Kingdom; [g]Leibniz Institute of Agricultural Development in Transition Economies, Germany; [h]University of Waterloo, Canada

The Complexities of Agent-Based Modeling Output Analysis

Journal of Artificial Societies and Social Simulation **18** (4) 4
<https://www.jasss.org/18/4/4.html>
DOI: 10.18564/jasss.2897

Received: 28-Apr-2015 Accepted: 29-Jun-2015 Published: 31-Oct-2015

Abstract

The proliferation of agent-based models (ABMs) in recent decades has motivated model practitioners to improve the transparency, replicability, and trust in results derived from ABMs. The complexity of ABMs has risen in stride with advances in computing power and resources, resulting in larger models with complex interactions and learning and whose outputs are often high-dimensional and require sophisticated analytical approaches. Similarly, the increasing use of data and dynamics in ABMs has further enhanced the complexity of their outputs. In this article, we offer an overview of the state-of-the-art approaches in analysing and reporting ABM outputs highlighting challenges and outstanding issues. In particular, we examine issues surrounding variance stability (in connection with determination of appropriate number of runs and hypothesis testing), sensitivity analysis, spatio-temporal analysis, visualization, and effective communication of all these to non-technical audiences, such as various stakeholders.

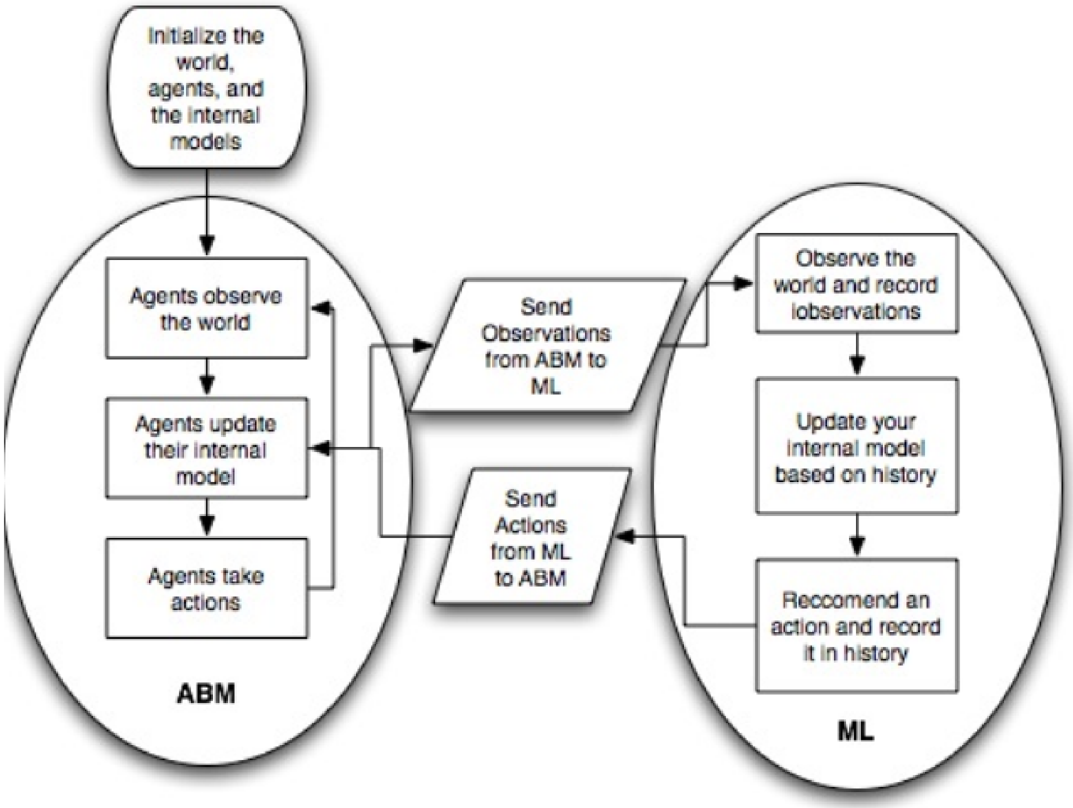
Hide
Article Tools
Smaller Text Bigger
Bookmark:
Export citation
Articles with the same author(s)
Articles with similar keywords
Articles citing this article
Download this article in PDF format
Alert me when this article is cited within JASSS
Send a link to this article to a colleague

Keywords:
Agent-Based Modelling, Methodologies, Statistical Test, Sensitivity Analysis, Spatio-Temporal Heterogeneity, Visualization

(Lee et al. 2015; Ten Broeke, Van Voorn, and Ligtenberg 2016)

Future of Agent-Based Modeling

- Developments in Machine Learning Models would affect ABM. (Rand 2006)



- Assignment:

- Go back to our PD simulation
- Keep only Randomer agent
- Give a an attribute `p` stands for probability of cooperating. This should be randomly picked from a distribution.
- Create an agent type `Regressor`
- The agent decides to cooperate or defect based on the logistic regression results. `from sklearn.linear_model import LogisticRegression` in order do exploit the opponent.

Next: Choosing a framework:

- You might not need a framework for all problems.
- If problems are visual and spatial, you might pick Netlogo.

- If you'd like to use external packages, you might pick Python based solutions.
- `AgentPy` is simple but growing.
-
- `Mesa` can be a good trade-off.
- If you are into Julia language, you can take a look at Agents.jl

Closing

- We've covered the basics of Agent-Based Modeling.
- We saw a small subset of examples. If you need inspiration, you can check Netlogo's Model Library.

Thanks!

References

Cariboni, J, D Gatelli, R Liska, and A Saltelli. 2007. "The Role of Sensitivity Analysis in Ecological Modelling." *Ecological Modelling* 203 (1-2): 167–82.

Downing, D Jo, RH Gardner, and FO Hoffman. 1985. "An Examination of Response-Surface Methodologies for Uncertainty Analysis in Assessment Models." *Technometrics* 27 (2): 151–63.

Lee, Ju-Sung, Tatiana Filatova, Arika Ligmann-Zielinska, Behrooz Hassani-Mahmooei, Forrest Stonedahl, Iris Lorscheid, Alexey Voinov, J Gareth Polhill, Zhanli Sun, and Dawn C Parker. 2015. "The Complexities of Agent-Based Modeling Output Analysis." *Journal of Artificial Societies and Social Simulation* 18 (4).

Rand, William. 2006. "Machine Learning Meets Agent-Based Modeling: When Not to Go to a Bar." In *Conference on Social Agents: Results and Prospects*.

Ten Broeke, Guus, George Van Voorn, and Arend Ligtenberg. 2016. "Which Sensitivity Analysis Method Should i Use for My Agent-Based Model?" *Journal of Artificial Societies and Social Simulation* 19 (1): 5.