The Journal of Open Source Software

Agents.jl: agent-based modeling framework in Julia

Ali R. Vahdati[1]

1 Department of Anthropology, University of Zurich, Zurich, Switzerland

DOI: 10.21105/joss.01611

Software

- Review
- Repository
- Archive

Submitted: 25 June 2019
Published: 08 October 2019

Summary

Agent-based modeling involves designing a system of autonomous agents that interact based on a set of given rules (Grimm et al., 2006). It is used for studying complex systems whose behavior cannot be easily identified using classical mathematical approaches.

Agent-based modeling provides a bottom-up approach for studying complex systems, whereas analytical models have a top-down one (Bonabeau, 2002). The two approaches are complementary and they both can benefit from insights that the other approach contributes. Analytical models make many simplifying assumptions about the systems they study. This results in systems that are tractable and lead to clear conclusions. Agent-based models, on the other hand, are more difficult to make sense of because they relax many assumptions of equation-based approaches. This is at the same time an advantage of agent-based models because it allows observing the effect of agent and environment heterogeneity and stochasticity, which can change a model's behavior (Farmer & Foley, 2009). Agent-based modeling is a particularly important tool for studying complex systems where a system's behavior cannot be predicted and has to be explored.

There are currently several agent-based modeling frameworks available (notable examples are NetLogo (Wilensky, 1999), Repast (North et al., 2013), MASON (Luke, Cioffi-Revilla, Panait, Sullivan, & Balan, 2005), and Mesa (Masad & Kazil, 2015)), but not for the Julia language, a new language designed with the needs of scientific computing in mind. Julia provides a combination of features that have historically been mutually exclusive. Specifically, languages that were fast to write, such as Python, were slow to run. And languages that were fast to run, such as C/C++, were slow to write. The combination of speed, expressiveness, and support for interactive computing makes Julia a desirable choice for scientific purposes. Agent-based models can involve hundreds of thousands of agents, each performing certain computations at each time step, thus speed is essential. Agents.jl is the first high-performance agent-based modeling framework to be written in Julia, and offers key advantages relative to the existing frameworks in other languages. First, unlike NetLogo, Agents.jl uses a general-purpose language rather than custom one, which reduces the learning curve and unifies the modeling and analysis language. Second, Julia is an easy to learn and easy to write language, unlike Java that is used for Repast and MASON, and provides a REPL (Read-Eval-Print-Loop) to build and analyze models interactively. Third, Julia is fast to run, unlike Python, which is used for Mesa (see Figure 1). This can be important in large agent-based models.

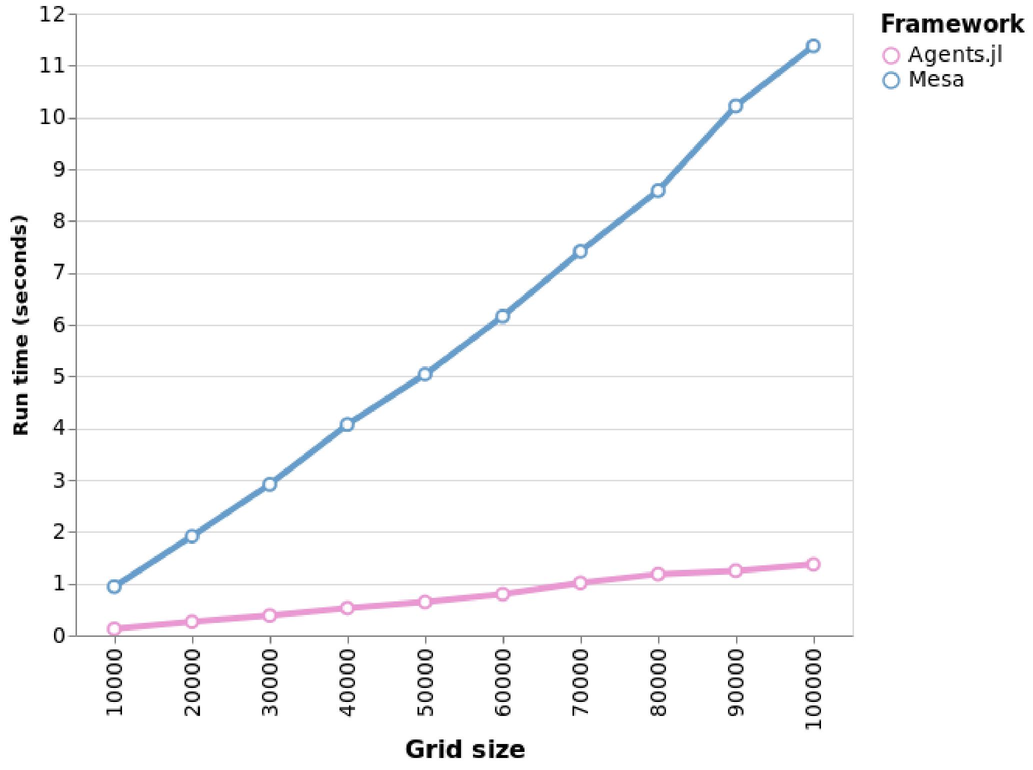

Figure 1: Speed comparison of a version of the "forest fire" model in Agents.jl vs Mesa. The same implementation of the model in Agents.jl (originally taken from Mesa's example and then re-implemented in Agents.jl) shows more than 8x speed gain. See the documentation for more details.

Agents.jl allows users to only think about what needs to happen during each step of their model, and the rest will be managed by the framework. Future development can include building a GUI to help users with less programming knowledge, and implementing real-time visualization of simulations. The documentation contains a tutorial and several example agent-based models to demonstrate the features and workflows supported by the package.

Features

- Built-in 2D and 3D regular grids with Moore and von Neumann neighborhoods and periodic edges.
- Automatic data collection into a "DataFrame".
- Automatic aggregation of collected data with user defined functions.
- Automatic aggregation of raw outputs with user-defined summary statistics.
- Interactive visualization of simulation outputs with "DataVoyager".
- Running and aggregating simulation replicates.
- Visualizing cellular automata.
- Parallel computation of simulation replicates.

References

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, *99*(Supplement 3), 7280–7287. doi:10.1073/pnas.082080899

Farmer, J. D., & Foley, D. (2009). The economy needs agent-based modelling. *Nature*, *460*(7256), 685–686. doi:10.1038/460685a

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., et al. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, *198*(1-2), 115–126. doi:10.1016/j.ecolmodel.2006.04.023

Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., & Balan, G. (2005). MASON: a multiagent simulation environment. *Simulation*, *81*(7), 517–527. doi:10.1177/0037549705058073

Masad, D., & Kazil, J. (2015). Mesa: an agent-based modeling framework. In *Proceedings of the 14th python in science conference* (pp. 53–60). SciPy. doi:10.25080/majora-7b98e3ed-009

North, M. J., Nicholson, T. C., Ozik, J., Tatara, E. R., Macal, C. M., Bragen, M., & Sydelko, P. (2013). Complex adaptive systems modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, *1*(3). doi:10.1186/2194-3206-1-3

Wilensky, U. (1999). NetLogo. Center for Connected Learning; Computer-Based Modeling, Northwestern University, Evanston, IL.