Game Theory and Python: An educational tutorial to game theory and repeated games using Python

Nikoleta E. Glynatsi[1,2] **and Vincent A. Knight**[1]

1 Department of Mathematics, Cardiff University, Senghennydd Rd, Cardiff CF24 4AG **2** Max Planck Institute for Evolutionary Biology, August-Thienemann-Straße 2, 24306 Plön

Summary

These materials are an open-source educational tutorial aimed at introducing participants to game theory and more specifically to repeated games. The tutorial uses the open-source library called `Axelrod-Python` and Jupyter Notebooks making this an open-source, reproducible and interactive tutorial.

Statement of Need

We are mathematicians and Python programmers with interests in game theory and pedagogy. This tutorial was created as a resource for computer workshops and for introducing mathematicians to programming. As a result, this tutorial is aimed at two groups of individuals:

- those familiar with Python (programmers) who want to start to learn game theory,
- mathematicians with little or no programming knowledge as a pathway to programming through the interesting subject that is game theory

Game theory is a field of applied mathematics interested in strategic interactions. Game theory itself has a number of subfields and the one considered in this tutorial is the area of repeated games. Repeated games have been the subject of research (Axelrod & Hamilton, 1981; Glynatsi & Knight, 2021; Nowak, 2006) but have also been used extensively as an entry to the subject for students at undergraduate level courses (Brokaw & Merz, 2004; Knight, 2015). This tutorial focuses on the Iterated Prisoner's Dilemma. The advantage of studying the Iterated Prisoner's Dilemma is that it models situations in which self-interest clashes with collective interest, thus it provides a framework for illustrating the usage of mathematics in real-life decision making.

Content

These materials consist of four tutorials that focus on the Iterated Prisoner's Dilemma, the notion of strategies and computer tournaments whilst using `Axelrod-Python` (The Axelrod project developers, 2016).

In 1980, a political scientist called Robert Axelrod ran a computer tournament of the Iterated Prisoner's Dilemma where strategies written in computer code would repeatedly

choose between self and collective interest. The strategies would decide on their next action using the history of previous interactions and the winner was decided based on the average score. The open-source package `Axelrod-Python` was created in order to reproduce Axelrod's original work but to also serve as an educational and research tool.

Following the tutorial, participants will be able to reproduce Axelrod's tournament but also create their own unique tournaments from a selection of more than 200 strategies. Furthermore, participants will be able to progress their comprehension of the topic by producing strategies and implement them in Python. The new strategies can then be placed in different tournaments in order to access their success and weakness. This progression of the tutorial fits within a constructive framework of learning (Jordan et al., 2008) and, by the nature of participants actively writing the required software, an active learning setting (Freeman et al., 2014). Participants with knowledge of Python will benefit from an understanding of game theory and participants with no knowledge of Python will also be introduced to basic programming concepts through the tangible ideas of the Prisoner's Dilemma.

Usage

The tutorial `Game Theory and Python` can be used in a workshop environment or through independent learning.

- **Workshop:** The material has been designed for a 2 hour workshop. A suggested timetable is:
  - 0:00 - 0:15 Installation (guidelines are given in the README)
  - 0:15 - 0:30 An introduction to game theory and the Iterated Prisoner's Dilemma
  - 0:30 - 0:55 Creating matches and tournaments using Axelrod-Python
  - 0:55 - 1:20 Writing strategies and contributing to Axelrod-Python
  - 1:20 - 1:50 Playing against strategies of the Iterated Prisoner's Dilemma
  - 1:50 - 2:00 Closing remarks and wrapping up

  In a workshop environment we suggest that the instructor has familiarized themselves with the written parts of the tutorial beforehand. The written parts of each notebook include an introduction to the topic being covered. It is advised that the instructor discusses the introduction to the topic followed by them typing out/running the material while the participants follow using their own machines. The instructor should encourage the participants to try the exercises of each notebook alone or with other participants. Before moving to the next notebook the instructor should encourage a discussion amongst everyone regarding the results of the exercises each had and their interpretation.

- **Independent Learning**: An independent learner should aim to spend the same amount of time on the material, approximately 2 hours. A suggested timetable is:
  - 0:00 - 0:15 Installation (guidelines are given in the README)
  - 0:15 - 0:30 An introduction to game theory and the Iterated Prisoner's Dilemma
  - 0:30 - 1:00 Creating matches and tournaments using Axelrod-Python
  - 1:00 - 1:30 Writing strategies and contributing to Axelrod-Python
  - 1:30 - 2:00 Playing against strategies of the Iterated Prisoner's Dilemma

  If the tutorial is being followed by an individual learner, we suggest that the learner reads the written parts of each notebook followed by running the tutorial and completing the exercises. The individual should take some time to reflect on the results of each notebook and their interpretation.

These instructions can also be found in the README. Though we recommend that the material can be covered in 2 hours, in both a workshop environment and through independent learning, some audience/learners might need more time to complete all the exercises. We recommend any facilitator/learner to adjust as necessary.

Recent Uses

This tutorial was originally formulated to run a game theory workshop at PyCon Namibia 2017. It has attracted the attention of the community ever since and has been used in other events.

Acknowledgements

References

Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. *Science*, *211*(4489), 1390–1396. https://doi.org/10.1126/science.7466396

Brokaw, A. J., & Merz, T. E. (2004). Active learning with monty hall in a game theory class. *The Journal of Economic Education*, *35*(3), 259–268. https://doi.org/10.3200/JECE.35.3.259-268

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, *111*(23), 8410–8415. https://doi.org/10.1073/pnas.1319030111

Glynatsi, N. E., & Knight, V. A. (2021). A bibliometric study of research topics, collaboration, and centrality in the iterated prisoner's dilemma. *Humanities and Social Sciences Communications*, *8*(1), 1–12. https://doi.org/10.1057/s41599-021-00718-9

Jordan, A., Carlile, O., & Stack, A. (2008). *Approaches to learning: A guide for teachers: A guide for educators*. McGraw-Hill Education (UK).

Knight, V. (2015). Playing games: A case study in active learning applied to game theory. *MSOR Connections*, *14*(1), 28–38. https://doi.org/10.21100/msor.v14i1.254

Nowak, M. A. (2006). Five rules for the evolution of cooperation. *Science*, *314*(5805), 1560–1563. https://doi.org/10.1126/science.1133755

The Axelrod project developers. (2016). *Axelrod: Version 4.3.0.* https://doi.org/10.5281/zenodo.1405868