The Journal of Open Source Software

Learning Simulator: A simulation software for associative learning

Markus Jonsson[1], **Stefano Ghirlanda**[1, 2], **Johan Lind**[1], **and Magnus Enquist**[1, 3]

1 Centre for Cultural Evolution, Stockholm University, Stockholm, Sweden **2** Department of Psychology, Brooklyn College and Graduate Center, CUNY, New York, NY, USA **3** Department of Zoology, Stockholm University, Sweden

DOI: 10.21105/joss.0XXXX

Software

- Review
- Repository
- Archive

Editor: Editor Name

Submitted: 01 January XXXX
Published: 01 January XXXX

Summary

Learning Simulator is a software for simulating learning phenomena governed by associative learning, applicable for example to animal learning experiments. It is primarily targeted to computational and behavior biologists, ethologists, and psychologists, however students/teachers who learn/teach learning phenomena may also find it useful.

Introduction

Learning Simulator was developed to study associative learning (AL) in animals, the process by which a subject learns contingency relations, either between pairs of stimuli (classical or Pavlovian conditioning), or between stimulus-behavior pairs (operant or instrumental conditioning) (Bouton, 2016; Pearce, 2013).

The simulator uses a commonly used framing of learning that comprises a subject interacting with an environment. The environment presents a stimulus to the subject, and the subject responds with a behavior. As a result, the environment presents the next stimulus that the subject responds to, and so on. See Figure 1.

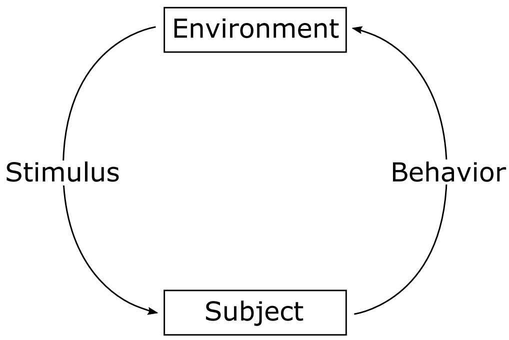

Figure 1: The subject and the world can be seen as two interacting dynamical systems, where the state variables in the subject determines the probabilities for its behaviors (the subject's output), and each behavior from the subject puts the environment in a state that determines its output stimulus.

The stimuli that the environment presents and the behaviors that the subject can exhibit are pre-defined by the user of the program. Each stimulus is given a reinforcement value (which is genetically determined for biological subjects). A rewarding stimulus (e.g. food) would typically have positive value, while a stimulus representing harm to the body ("punishment") would have a negative value.

As seen in Figure 1, the consequence of responding with a behavior (say B) to a stimulus (say S) is that the subject meets the next stimulus (say S'):

$$S \to B \to S'.$$

The reinforcement value of S' gives the subject an indication of the quality of the response B to S. Specifically, this is accomplished by updating one or more of the subject's memory state variables. The values of these state variables control the probabilities of future responses: if the response B to stimulus S leads to a reward (a stimulus with high reinforcement value), the subject will be more likely to respond with B the next time it faces S.

The user of Learning Simulator specifies in a text-based script how the output stimulus from the environment depends on the subject's response to the previous stimulus. This script also specifies the values of all parameters used in the learning process. The simulation script, written in a simple and well-documented scripting language, is the only input to Learning Simulator. The user also specifies how to visualize the simulation data, for example how a memory state variable changes over time during the simulation. Learning Simulator also includes a functionality to export the results to a data processor spreadsheet.

More information is available at https://www.learningsimulator.org.

Applications of associative learning

Associative learning theory has a rich tradition of computational modeling. During the last decade or so, AL has proven increasingly powerful, as a fair amount of research in the field has been directed toward the development of different mathematical models, *learning mechanisms*.

Firstly, when applied to deep neural networks, AL has been used to teach computers to find optimal play and achieve human level skills in chess (Silver et al., 2017) and the Chinese board game Go (Silver et al., 2016).

Secondly, behaving optimally (or near-optimally) is central to animals' adaptation to their environment. Thus, AL can also provide explanations for a wide range of learning phenomena in both human and non-human animals (Enquist et al., 2016; Ghirlanda et al., 2020). This also enables the possibility of generating predictions of animal behavior.

Moreover, AL theory underpins some of the most successful applications of psychology to animal welfare and training (McGreevy & Boakes, 2011). It has also proven important for applications related to human health (Bernstein, 1999; Haselgrove & Hogarth, 2013; Schachtman & Reilly, 2011).

The ability of AL algorithms to be able to search for optimal policies using low-variance gradient estimates has made them useful in several other real-life applications, such as robotics, power control, and finance (Grondman et al., 2012).

Statement of need

As a result of the many application areas of AL, there is now a plethora of mechanisms with varying properties and varying predictive power in different environments.

This has given rise to a need for a general simulation software for simulating different AL mechanisms. The first aim of our software is to fulfil this need.

The Journal of Open Source Software

The second aim is to provide a generic, flexible way to describe very different animal learning experiment trial structures.

The main advantage of Learning Simulator is its simple scripting language that provides a way to explore/understand different learning mechanisms and investigate the effects of varying their underlying parameters.

Another strength of Learning Simulator lies in the simplicity to specify even complex environments with which the subject interacts, for example an experiment trial structure. The scripting language has been developed to be available to any researcher of learning phenomena – not necessarily computer programmers.

Our software has been used in scientific publications (Ghirlanda et al., 2020; Lind, 2018; Lind et al., 2019) as well as in teaching, both at the Ethology Master's Programme at Stockholm University, and at the Veterinary Programme at the Swedish University of Agricultural Sciences.

An open source license as well as its accessibility enables further scientific exploration of learning phenomena by students and experts alike within the fields of biology, ethology, and psychology.

State of the field

Other simulating software either specialize in one specific mechanism (Schultheis et al., 2008a, 2008b; Alonso et al., 2012) or only includes models of classical conditioning (George, 2019; Harris & Livesey, 2010; Thorwart et al., 2009), or where both the mechanism and environment are hard-wired (*Q-Learning Simulator*, n.d.).

Learning Simulator includes several mechanisms of AL in a common program platform:

- *Stimulus-response learning* (Bush & Mosteller, 1951),
- *Q-learning* (C. J. C. H. Watkins, 1989; C. J. Watkins & Dayan, 1992),
- *Expected SARSA* (Van Seijen et al., 2009),
- *Actor-critic* (Witten, 1977),
- *A-learning* (Ghirlanda et al., 2020), and
- *Rescorla-Wagner* (Wagner & Rescorla, 1972),

facilitating direct comparison of these mechanisms. Moreover, the flexible environment definition allows the generation of meaningful experiment designs and discrimination tasks.

Repository

The program is written in Python and its source code repository is hosted on GitHub. Its documentation is hosted on Read the Docs. The repository incorporates continuous integration with automatic build of the documentation, linting, and execution of a test suite along with code coverage measurement.

Acknowledgements

Research and development was supported by the Knut and Alice Wallenberg Foundation (2015.0005).

We acknowledge valuable contributions from Vera Vinken during the development of this project.

References

Alonso, E., Mondragón, E., & Fernández, A. (2012). A java simulator of rescorla and wagner's prediction error model and configural cue extensions. *Computer Methods and Programs in Biomedicine*, *108*(1), 346–355.

Bernstein, I. L. (1999). Taste aversion learning: A contemporary perspective. *Nutrition*, *15*(3), 229–234.

Bouton, M. (2016). *Learning and behavior: A modern synthesis*. Sunderland, MA: Sinauer.

Bush, R. R., & Mosteller, F. (1951). A mathematical model for simple learning. *Psychological Review*, *58*(5), 313.

Enquist, M., Lind, J., & Ghirlanda, S. (2016). The power of associative learning and the ontogeny of optimal behaviour. *Royal Society Open Science*, *3*(11), 160734.

George, D. N. (2019). In *GitHub repository*. GitHub. https://github.com/DavidNGeorge/learnSim

Ghirlanda, S., Lind, J., & Enquist, M. (2020). A-learning: A new formulation of associative learning theory. *Psychonomic Bulletin & Review*, 1–29.

Grondman, I., Busoniu, L., Lopes, G. A., & Babuska, R. (2012). A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 1291–1307.

Harris, J. A., & Livesey, E. J. (2010). An attention-modulated associative network. *Learning & Behavior*, *38*(1), 1–26.

Haselgrove, M., & Hogarth, L. (2013). *Clinical applications of learning theory*. Psychology Press.

Lind, J. (2018). What can associative learning do for planning? *Royal Society Open Science*, *5*(11), 180778.

Lind, J., Ghirlanda, S., & Enquist, M. (2019). Social learning through associative processes: A computational theory. *Royal Society Open Science*, *6*(3), 181777.

McGreevy, P., & Boakes, R. (2011). *Carrots and sticks: Principles of animal training*. Darlington Press.

Pearce, J. M. (2013). *Animal learning and cognition: An introduction*. Psychology Press.

Q-learning simulator. (n.d.). https://www.mladdict.com/q-learning-simulator.

Schachtman, T. R., & Reilly, S. S. (2011). *Associative learning and conditioning theory: Human and non-human applications*. OUP USA.

Schultheis, H., Thorwart, A., & Lachnit, H. (2008a). HMS: A MATLAB simulator of the harris model of associative learning. *Behavior Research Methods*, *40*(2), 442–449.

Schultheis, H., Thorwart, A., & Lachnit, H. (2008b). Rapid-REM: A MATLAB simulator of the replaced-elements model. *Behavior Research Methods*, *40*(2), 435–441.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., & others. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., & others. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv Preprint arXiv:1712.01815*.

Thorwart, A., Schultheis, H., König, S., & Lachnit, H. (2009). ALTSim: A MATLAB simulator for current associative learning theories. *Behavior Research Methods*, *41*(1), 29–34.

Van Seijen, H., Van Hasselt, H., Whiteson, S., & Wiering, M. (2009). A theoretical and empirical analysis of expected sarsa. *2009 Ieee Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 177–184.

Wagner, A., & Rescorla, R. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical Conditioning Ii*, *64*, 99.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3-4), 279–292.

Witten, I. H. (1977). An adaptive optimal controller for discrete-time markov environments. *Inf. Control.*, *34*(4), 286–295.