# Guide to the learning simulation lab at the Centre for Cultural Evolution

Guide used on advanced course
"Human behaviour: biology and culture" at
Department of Zoology, Stockholm University

Johan Lind & Markus Jonsson

February 12, 2020

Centre for Cultural Evolution, Stockholm University,
Lilla Frescativägen 7, 106 91 Stockholm, Sweden.
johan.lind@su.se, markus.jonsson@su.se

## Introduction

This is a guide that showcases associative learning. The guide contains examples and questions. The aim is not to give a complete picture of associative learning, but to introduce some key concepts and processes, illustrate factors that affect learning of both single behavior and behavior sequences, and to show some examples of the power of associative learning. For a more complete account of learning, we recommend Bouton's textbook *Learning and behavior* [2].

The scripts contained in the text-files are to be used in a computer program that has been made to simulate learning phenomena. In short, the simulations are based on defining a world and parameters for an organism that is subjected to that world, following the formalism described in *The power of associative learning and the ontogeny of optimal behaviour* [3]. The world is defined according to what stimuli are present and what the consequences are when subjected to these stimuli. The organism is defined according to what behaviors it can perform, what its initial conditions are and parameters that determine memory updates and exploration. An underlying assumption is that behaviours are chosen that maximize the total value collected from the world in which it operates. See Appendix for equations and brief model details.

The simulator can produce both figures and data. Simulations can be performed with different learning mechanisms. Simulations below are prepared for using both stimulus-response learning (`SR` in the scripts), and a combination of

stimulus-response- and stimulus value learning. This latter mechanism combines instrumental and Pavlovian conditioning, and is called `GA` in the scripts[1].

All necessary software has been pre-installed, but for those interested the learning simulator can be downloaded at:

`https://github.com/markusrobertjonsson/lesim2.`

Running the learning simulator requires Python 3.6 (`https://www.python.org/downloads/`) and Matplotlib (`https://matplotlib.org/`). With Python and Matplotlib installed, the file `lesim.py` opens the learning simulator where scripts can be used. For details about the learning simulator and how to run the scripts see the user guide 'ug.pdf' that is included in the learning simulator package.

## Review the concepts and parameters

Before starting the simulations, review the following concepts and parameters carefully:

**SR-mechanism** Of the three mechanisms used in these simulations, `SR` refers to stimulus-response learning. This only includes updates of one kind of memory, that is the stimulus-response assocations. This value is referred to as memory $v$ of the associative strength between stimulus $S$ and behavior $B$. This is updated after behavior $B$ has been performed towards stimulus $S$ with the consequence $S'$, or after the following event sequence $S \rightarrow B \rightarrow S'$. $S'$ is a stimulus that has some kind of primary value, so that it can act as a reinforcer.

**GA-mechanism** The second mechanism used in the simulation. `GA` refers to genetically guided associative learning and was formalized in Enquist et al. [3]. `GA` includes both learning about stimuli (Pavlovian learning) and learning about actions (instrumental learning, or trial-and-error learning as it is also called). The instrumental part comes from the stimulus-response associations ($v$). Here the mechanism can learn about actions from their consequences. The Pavlovian part comes from learning about stimulus values ($w$). This is a memory for the value of stimulus $S$. It is updated as a consequence of stimulus $S'$ coming after stimulus $S$. It is not affected by what response $B$ was performed. The stimulus value $w$ is the same as the related concept conditioned reinforcement in experimental psychology, and secondary reinforcement in animal training [5].

**Rescorla-Wagner** A classic model for Pavlovian conditioning. Here, an association between a conditioned and an unconditioned stimulus is learned. In the scripts below, this association is called *vss* (associative strength between the two stimuli). In contrast to the mechanisms above, the value of the unconditioned stimulus is called `lambda.`

[1] '`GA`' is short for genetically guided associative learning.

**Initial values** An organism will always enter new situations with some initial values. Initial values can vary from an individual that is completely naïve to an individual that has a lot of experiences. Initial values can affect the rate of learning a lot, and in the scripts the are called `'start_v'`.

**Learning rate** Parameters $\alpha_v$and $\alpha_w$ in the model regulate the rate at which the two memories are updated. These are called `'alpha_v'` and `'alpha_w'` in the scripts.

**Exploration** The $\beta$ parameter is part of the decision making mechanism and it regulates the amount of exploration. If $\beta = 0$ all behaviors are equally likely to be chosen. This means that prior learning has no effect on the decision made. If $\beta$ is large the behavior with the highest $v$-value is will be selected with appreciable probability. This means that prior learning has a greater effect on decisions.

**Primary reinforcer** It is assumed that primary values have been set by genetic evolution to reflect a stimulus' direct contribution to the animal's survival and reproduction. These can be negative for stimuli that are painful or distasteful, or positive for stimuli that are beneficial for animals, such as food for a hungry animal, water for a thirsty animal, or perhaps the presence of a conspecific or relative for a young or a social animal. The primary value of $S$ is written $u(S)$ and in the scripts these are written as `'u'`.

**Question:**

- Before starting the simulations, think about and discuss briefly the relationship between learning and evolution through natural selection. What factors of the learning model can be genetically determined?

# 1 Simulations

All scripts are included in separate text files. The name of the text files starts with the section names. To run a script, first open the learning simulator by double clicking the file `lesim.py`. Now, open the file from the file menu in the learning simulator, and click the `Simulate and Plot` button in the lower left corner. To close all graph windows, use the button in the lower right hand corner that says `Close All Figures`. The text is fully editable within the learning simulator.

## 1.1 Instrumental: One individual learning a single response

**File to open:** `1-1-instrumental.txt`

1. Read the script, note what behaviors are included and what stimuli behaviors can be directed to.

2. Run the script (click the `Simulate and Plot` button in the lower left corner). This produces `Figure 1`.

3. Repeat step 2 several times. Note that the probability of lever pressing varies due to the stochastic nature of the decision making function. Repeat step 2 until you get a picture of the variation of how many trials it takes to learn lever pressing under these circumstances.

4. Clear the screen by pressing the `Close All Figures` button.

5. Remove the hash `(#)` in front of the last four lines and run the script again. This illustrates the underlying changes to memory.

6. Repeat step 2 again, several times, and note the relationship between the changes in memory (associative strength between the behavior `Press` to the stimulus `Lever`) and the probability of lever pressing.

**Questions:**

- What causes variation in the speed of learning in this scenario?
- What causes responding to the lever?
- What memory is changed in this simulation?

## 1.2 Instrumental learning: A group of individuals learning a single response

**File to open:** `1-2-Intrumental-groups.txt`

1. Read the script. This script describes the same situation as the previous script. But, as indicated on line `'subjects',` more than one individual can be included in the script.

2. Run the script to produce a figure. This figure shows five individual runs.

3. Remove the hash from the two last lines (`@figure 'Probability of responding'` and the line below).

4. Run the script to produce a figure. This figure shows the average for the five individual runs.

5. Repeat step 4 several times.

**Questions:**

- What differences can you detect in learning speed when comparing single individual trajectories with group averages?
- Discuss what consequences this can have for interpretations of individual differences in learning speed.

## 1.3 Instrumental learning: Explore parameters in a group of individuals learning a single response

**File to open:** `1-3-Intrumental-groups.txt`

1. Read the script. As indicated on line `'subjects',` many individuals can be included in a script. Now 100 individual runs are included in the script.
2. Run the script to produce a figure.
3. Now explore how changing parameter values changes the rate of learning a single behavior. Change one parameter at a time. Explore the following paramaters:
   (a) Food reward value (`u),` change to e.g. 2, 10, 100.
   (b) Behavior cost, change to e.g. 0, 5, 15.
   (c) Exploration (`beta`), change to e.g. 0,1,3.
   (d) Learning rate (`alpha_v`), change to e.g. 0.1, 0.9.

   When finished having explored one parameter, it can be helpful to both clear all figure windows, and revert to these initial settings before exploring the next parameter:

```
alpha_v          : 0.1
beta             : 1
behavior_cost    : press:0, default:0
u                : food:10, default: 0
```

**Questions:**

- What parameters determine how fast learning is?
- Can very rapid learning be bad for an individual? Discuss pros and cons with rapid and slow learning.

## 1.4 Pavlonian conditioning

**File to open:** `1-4-Pavlovs-dog.txt`

1. Read the script. This is a standard case of Pavlonian conditioning where a conditioned stimulus (CS) and a biologically relevant unconditioned stimulus (US) are paired. Initially, the CS will not cause a conditioned respons (CR), but the US will release an unconditioned response. After multiple pairings of US and CS, a conditioned response (CR) develops towards the CS. In the script, `us` is the unconditioned stimulus (food), `cs` is the conditioned stimulus (sound of a bell), and `cr` is the conditioned response (salivate when hearing the bell). Note the acquisition of the stimulus-stimulus associations.

2. Vary `us` (try negative values as well) and note change in learning speed.

3. Vary the `beta`-parameter and note what happens.

   When finished having explored one parameter, it can be helpful to both clear all figure windows, and revert to these initial settings before exploring the next parameter:

```
start_vss          : default:0
alpha_vss          : 0.1
beta               : 1.2
behavior_cost      : default:0
lambda             : us:10, default: 0
```

**Questions:**

- What determines learning speed?
- Why was learning not affected by change in the beta-parameter?

## 1.5 Chaining - learning behavior sequences

**File to open:** `1-5-Chaining.txt`

For chaining to be possible, neutral stimuli (or CS in Pavlovian terms) must acquire value. This way, a previously neutral stimulus can, after having acquired some value ($w \neq 0$), act as a reinforcer. In contrast to primary reinforcers (stimuli with $u > 0$), initially neutral stimuli have no primary reinforcement value, $u = 0$).

In this excercise a behavior sequence will be learned. The excercise starts with showing that behavior sequences cannot be learned without a mechanism that can learn that neutral stimuli can be rewarding. For this reason this exercise start with using the `SR` mechanism. This only allows learning about the value of performing a behavior towards a stimulus. In other words, the `SR` mechanism only includes updates of the stimulus-response association ($v$). When changing to the `GA` mechanism in step 3, both updates of stimulus-response associations ($v$) and stimulus values ($w$) are allowed.

1. Read the script. Note that we first start with using stimulus-response learning only (`mechanism : SR`). This prevents the acquisition of stimulus values ($w$).

   In this scenario, an animal can learn that eating a fruit is rewarding. The animal can also learn that fruits are present in trees, thereby learning to approach trees to find fruits. In this scenario the animal will encounter trees, which makes it possible to find a fruit in the tree (look at detail in `@phase`). It is rewarding to eat the fruit. The animal will rapidly learn to eat fruit. The final behavior sequence that can be learned is as follows:

   $S_{\text{Tree}} \rightarrow B_{\text{Approach}} \rightarrow S_{\text{Fruit}} \rightarrow B_{\text{Eat}} \rightarrow S_{\text{Reward}}$

2. Run the script. Look at both figures. Note what is learned and what is not learned.

3. Now change the mechanism used in the simulation, change from `mechanism : SR` to `mechanism : GA`. Now, both stimulus-response ($v$) and stimulus values ($w$) are included. Previously neutral stimuli can now acquire value, and thereby act reinforcing in their own right (see Appendix for brief details of the model and differences between `'SR'` and `'GA'`). Run the script again.

4. Compare the output of the two different mechanisms, both in terms of productive behavior and what behaviors are learned in the two cases.

5. Explore variation in parameter values to see how learning of behavior sequences can speed up. When finished having explored one parameter, it is advisable to revert to initial settings before exploring the next parameter:

```
alpha_v          : 0.1
alpha_w          : 0.1
beta             : 1
u                : reward:5, default: 0
```

**Questions:**

- What changed when the mechanism was changed from `SR` to `GA`?
- Why is it not possible to learn approach behavior towards the tree with only stimulus-response learning?
- What stimulus acquired value to reinforce the behavior `approach`?
- Can you find three ways to speed up the learning of a behavior sequence? What is most efficient?

## 1.6 An example of social learning

**File to open:** `1-6-Responding-to-nonsocial-stimulus.txt`

Here is an example where social learning is compared with individual learning. Here two different phases in the script are compared, one phase covers the scenario with only individual learning and the other phase describes a case with social learning.

In this scenario let the animal be somewhat neophobic to new food. This will be represented in the script as a positive value for being passive when subjected to a new kind of food, such as a fruit , or $v_{\text{fruit}\rightarrow\text{ignore}}$=4.

1. Read the script and focus first on the first phase, the one labeled `'individual learning'`. In this first scenario, the animal finds a tree with a fruit. In this first run, the animal is not passive towards the fruit so $v_{\text{fruit}\rightarrow\text{ignore}}$=0.

Run the script and look at the rate of learning to eat the fruit (ignore the empty lower half of the graph).

2. Examine the script. To make the animal neophobic towards new food, change `'start_v'` for being passive towards the fruit from 0 to 4. Or, change `start_v : Fruit->ignore:0` to `start_v : Fruit->ignore:4`

3. Run the script again and note the change in speed of learning when a stimulus is initially ignored.

4. Look at the phase labeled `'Social_learning'`. Here the animal will end up in the tree with a parent with the probability of 0.2. If the animal is in the presence of a parent it is more likely to try new food. This is represented in the script as `start_v : Parent->eat:10.` This initial value $v_{\text{Parent}\rightarrow\text{Eat}}$=10 will make the animal more likely to try to eat things when a parent is present.

5. Remove the first two triplet hash signs. Run the script (keep $v_{\text{fruit}\rightarrow\text{ignore}}$=4) and look at the probability plot, comparing individual and social learning.

6. Remove the next two triplet hash signs. Run the script and look both at the probability plot and the plot of stimulus-response values.

7. Remove the last two triplet hash signs. Run the script and look both at the plot of stimulus-response- and the stimulus values.

8. Change initial values (`'start_v'`) of $v_{\text{Fruit}\rightarrow\text{Passive}}$ and $v_{\text{Parent}\rightarrow\text{Eat}}$ and explore the importance of initial values for social learning.

   Copy and paste from below to revert to initial settings before exploring the next parameter:

```
start_v   : Fruit->ignore:4, Parent->eat:10, default:0
alpha_v   : 0.1
alpha_w   : 0.1
beta      : 1
behavior_cost : default: 0
u         : reward:25,pass_value:4,Parent:10,default: 0
```

**Questions:**

- Why was learning slower when changing $v_{\text{Fruit}\rightarrow\text{Passive}}$=0 to $v_{\text{Fruit}\rightarrow\text{Passive}}$=4?
- In what way did social learning speed up learning to eat the fruit?
- This is just one possibility whereby a social stimulus can result in efficient social learning. Discuss more ways in which social learning through associative processes can speed up learning new behavior.
- Did you find settings of initial values that were more or less favourable for social learning of new behaviors?

### 1.7 Bonus material - A case of chimpanzee nutcracking

The script `1-7-Nut-cracking.txt` describes a way to simulate the development of a longer behavior sequence. If you have time left, run the script, look at the output, modify parameters and explore the script any way you want.

This script is modelled after Inoue-Nakamura and Matsuzawa's description of how chimpanzees learn to open nuts with a stone tool [4]. This script is similar, but not identical, to the simulation of nutcracking behavior in *The power of associative learning and the ontogeny of optimal behaviour* [3].

- From empirical data we know that inexperienced chimpanzees steal open nuts from their mothers. You see two different phases. Note the differences in the fourth column.
  - `@phase with_open_nuts`: Here, there is a small chance ($p = 2/100$) that the chimpanzee goes directly to the state seeing an open nut.
  - `@phase no_open_nuts`: Here, all steps must be experienced from start to end.
- Run the script (`@phase with_open_nuts`).
- When you have first tried `@phase with_open_nuts` you can try to run the other phase. Add and delete hash signs to vary which run-statement you want to use.

**Questions:**

- One idea here is that a young chimpanzee that explores the situation next to an experienced tool using mother may encounter the different states by chance. What effect does the presence of the experienced mother have on the learning of tool use for young chimpanzees? Can you think of more than one advantage of the presence of the experienced mother?
- Is this a relevant description of the development of nutcracking behavior in chimpanzees?
- What can we learn from trying to simulate long behavior sequences observed in the field?

## References


[1] D. S Blough. Steady state data and a quantitative model of operant generalization and discrimination. *Journal of Experimental Psychology: Animal Behavior Processes*, 104(1):3–21, 1975.

[2] M. E. Bouton. *Learning and behavior: A contemporary synthesis*. Sinauer, 2nd edition, 2016.

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

[4] N. Inoue-Nakamura and T. Matsuzawa. Development of Stone Tool Use by Wild Chimpanzees (*Pan troglodytes*). *Journal of Comparative Psychology*, 11(2):159–173, 1997.

[5] Paul McGreevy and Robert Boakes. *Carrots and sticks: Principles of animal training*. Darlington Press, 2011.

[6] R. A. Rescorla and A. R Wagner. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In *Classical conditioning: current research and theory*. Appleton-Century-Crofts, 1972.

# Appendix

Here follows a brief description of the learning model [3]. Note that the mechanism called `'GA'` in the scripts contains updates of both stimulus-response associations ($v$) and stimulus values ($w$), whereas the mechanism called `'SR'` in the scripts only contains updates of stimulus-response associations ($v$).

An animal has a behavior repertoire and it can use its behaviors to navigate in a world of detectable environmental states. A behavior takes the animal from one state to another. Each state, or stimuli, has a primary reinforcement value that is genetically fixed. These values can be negative, neutral, or positive, and they guide learning so that behaviors favoring survival and reproduction are promoted. Animals are assumed to make choices that maximize the total value, and expectations of the value of a future state can develop [section 2.3. in 3].

In short, the model describes learning of sequences of behavior towards stimuli through changes in memory. It includes decision making that takes memory into account to determine what behavior should be selected when a given stimulus is perceived. Take for instance learning a single behavior, such as when a dog learns to give its paw in response to the command 'shake'. Lifting the paw is the behavior, the command 'shake' and the reward are stimuli. The event sequence to be learned is: command 'shake' →lift paw →reward, or

$$S_{\text{command 'shake'}} \rightarrow B_{\text{lift paw}} \rightarrow S_{\text{food reward}}$$

The model collects information about the value of performing behaviors towards different stimuli (or states), and information about the value of different stimuli (or being in specific states). Learning occurs through updates of two different kinds of memories. These memories correspond to Pavlovian and instrumental learning and are updated after an event sequence like in the dog example, or in general terms the event sequence $S \rightarrow B \rightarrow S'$. The first kind of memory is a stimulus-response association. $v_{S \rightarrow B}$ is used to denote the associative strength between stimulus $S$ and behavior $B$. In functional terms,

$v_{S \to B}$ can be described as the estimated value of performing behavior $B$ when perceiving stimulus $S$. The second memory stores the value of a stimulus. $w_S$ is used to denote the stimulus value and it is updated according to the value of a subsequent stimulus. In other words $w_S$ is the conditioned reinforcement value of being in state $S$. These memories are updated according to:

$$\begin{cases} \Delta v_{S \to B} = \alpha_v (u_{S'} + w_{S'} - v_{S \to B}) \\ \Delta w_S = \alpha_w (u_{S'} + w_{S'} - w_S) \end{cases} \tag{1}$$

after experiencing the event sequence $S \to B \to S'$. The stimulus-response association $v_{S \to B}$ is updated according to $u_{S'}$ a primary inborn fixed value of stimulus $S'$, and $w_{S'}$ the conditioned reinforcement value and the previously stored stimulus-response association $v_{S \to B}$. With conditioned reinforcement, the value of performing behavior $B$ when perceiving stimulus $S$ is the sum of the primary and conditioned reinforcement value of stimulus $S'$. If only the first equation is used and $w$ is excluded, then it represents instrumental stimulus-response learning, that is an instrumental version of the classic Rescorla-Wagner learning model [6, 1]. The learning rates $\alpha_v$ and $\alpha_w$ determine the rate at which memory updates take place.

For the learning model to generate and select behavior a mechanism for decision making is needed. A decision making mechanism was chosen that selects behavioral responses and causes some variation in behavior through exploration. This specifies the probability of behavior $B$ in state $S$ as:

$$\Pr(S \to B) = \frac{\exp(\beta v_{S \to B})}{\sum_{B'} \exp(\beta v_{S \to B'})} \tag{2}$$

that includes a parameter $\beta$ that regulates the amount of exploration. All behaviors are equally likely to be selected if $\beta = 0$ without taking estimated values into account. If $\beta$ is large, then the behavior with the highest estimated value ($v$) will mainly be selected.

Let us return to the dog for a practical example. The dog hears the command 'shake', stimulus $S$. If the dog moves its paw upwards, that is performing behavior $B$, it will receive the reward $S'$. The food reward $S'$ has a primary inborn value $u$. When the dog receives this reward after having responded correctly to the command 'shake', the stimulus-response memory $v_{\text{command 'shake'} \to \text{lift paw}}$ will increase according to the top row in equation 1. In addition, the stimulus value $w$ of the command 'shake' will be updated according to the bottom row of equation 1. This value $w$ of command 'shake' will approach the value $u$ of the food reward, and thereby gain reinforcing properties in its own right; it has become a conditioned reinforcer. The conditioned reinforcer can pave the way for learning more behaviors before moving the paw upwards. This can happen because behaviors that result in the dog hearing the command 'shake' can be reinforced.