City Research Online

City St George's, University of London

Citation: Mondragon, E., Gray, J. & Alonso, E. (2013). A Complete Serial Compound Temporal Difference Simulator for Compound stimuli, Configural cues and Context representation. Neuroinformatics, 11(2), pp. 259-261. doi: 10.1007/s12021-012-9172-z

This is the accepted version of the paper.

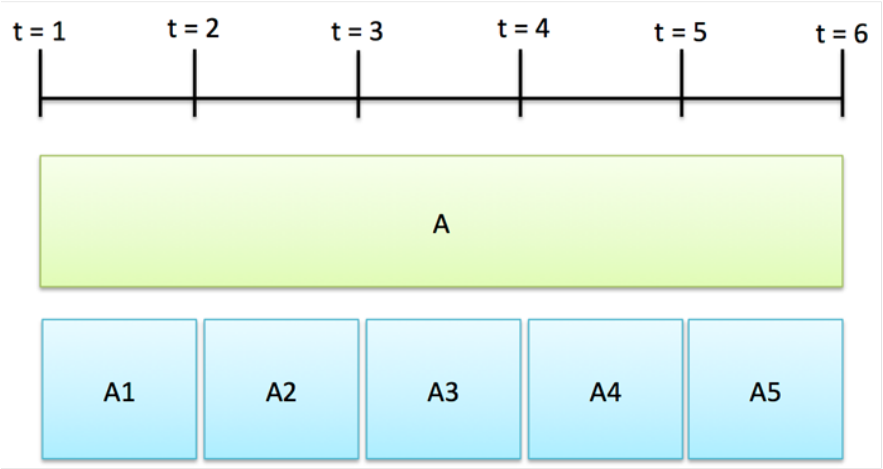
This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

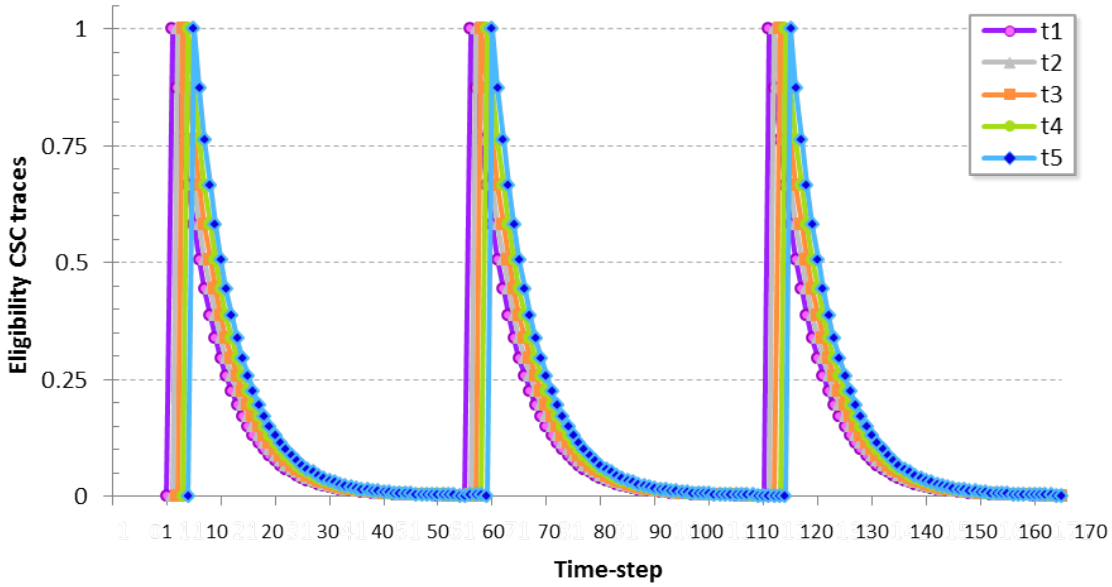
Permanent repository link: https://openaccess.city.ac.uk/id/eprint/5183/

Link to published version: https://doi.org/10.1007/s12021-012-9172-z

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk


t = 1
t = 2
t = 3
t = 4
t = 5
t = 6
A
A1
A2
A3
A4
A5


Eligibility CSC traces
1
0.75
0.5
0.25
0
1
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
Time-step
t1
t2
t3
t4
t5

```
let prediction = 0
for each time-step in the trial
   lastPrediction = prediction
   prediction = 0
   for all stimulus
     if the stimulus is active then
       set the next component of it to active
       prediction = prediction + the associative strength of the active component
   if prediction < 0 then
     prediction = 0
   if lastPrediction < 0 then
     lastPrediction = 0
   let predictionError = gamma* prediction — lastPrediction
   if the US is active then
     predictionError = predictionError + US asymptote
   let betaError = predictionError* beta
   for each stimulus
     for each component
       update associative strength by alpha* betaError* trace
       update the trace
for all stimuli
   store the results
   make next component the first component
```

A Complete Serial Compound (CSC) Temporal Difference Simulator

Esther Mondragón[1], Jonathan Gray[2], Eduardo Alonso[*2]

[1]Centre for Computational and Animal Learning Research, St Albans, UK; [2]Department of Computing, City University London, London, UK

[*]Corresponding author. E-mail: E.Alonso@city.ac.uk

Temporal Difference (TD) (Sutton and Barto, 1987[1]) is a real-time error correction model in which learning is computed according to the difference between successive predictions and a discount factor that decays exponentially, reflecting the fact that predictors closer to a reinforcer (the unconditioned stimulus, US) are based on more recent information and thus more accurate. In addition, an eligibility trace modulates the extent to which the stimulus predictive value is susceptible of changing on any given time-step.

The way stimuli are represented affects significantly how learning is implemented in TD. The Complete Serial Compound representation (CSC) (Moore et al., 1998[2]) has become standard in studies of dopamine function (Schultz 2010[3]) and is central in investigating reward-based models of schizophrenia (Smith et al., 2006[4]). CSC TD assumes that a stimulus can be broken down into a series of individual elements, which are each active for a single unit of time as shown Figure 1, left.

FIGURE 1 ABOUT HERE

Figure 1. Left: stimulus temporal representation according to CSC TD. Stimulus A is mapped into independent components {A1, A2, …, A6}, one per time-step {t1, t2, …, t6}. Right: eligibility traces per component and time-step.

Each of these new elements has a separate eligibility trace and associative strength, receives distinct reinforcement from the US, and while active contributes to the prediction term. In essence CSC TD treats the components of a stimulus as unique stimuli in their own right identifiable by the overarching stimulus and their position in the sequence. These component stimuli are linked only in that their activation is contingent on the activation of the supra-stimulus and their position in the sequence of time-steps. Briefly, the first component of the CSC stimulus becomes active when the supra-stimulus becomes active, and the active state of each component in the sequence is then the product of the active state of the preceding stimulus and that of the supra-stimulus. This representation produces correct predictions at the intra-trial level, because elements of the stimulus occurring distantly from the US receive correspondingly less reinforcement, modulated by their eligibility trace as shown in Figure1, right.

CSC TD is expressed formally as follows:

---

[1] Sutton, R.S., & Barto, A.G. (1987). A temporal-difference model of classical conditioning. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, pp. 355-378.

[2] Moore, J., Choi, J., & Brunzell, D. (1998). Predictive timing under temporal uncertainty: the TD model of the conditioned response. In D. Rosenbaum & A. Collyer (Eds.), *Timing of Behavior: Neural, Computational, and Psychological Perspectives* (pp. 3-34). Cambridge, MA: MIT Press.

[3] Schultz, W. (2010). Dopamine signals for reward value and risk: basic and recent data. *Behavioral and Brain Functions*, *6*, 6-24.

[4] Smith, A., Li, M., Becker, S., & Kapur, S. (2006). Dopamine, prediction error, and associative learning: a model-based account. *Network: Computation in Neural Systems*, *17*, 61–84.

$$V_{i,j}(t+1) = V_{i,j}(t) + \beta(X_{0,t+1} + \gamma P(t+1) - P(t))\alpha_i \bar{X}_{i,j}(t) \quad (1)$$

$$P(t) = \sum_i \sum_j (X_{i,j}(t)\, V_{i,j}(t)), if\ P(t) \geq 0, else\ 0 \quad (2)$$

$$X_{i,j}(t) = \begin{cases} 1\ if\ the\ jth\ element\ of\ the\ ith\ stimulus\ is\ present\ at\ time\ step\ t \\ otherwise\ 0 \end{cases} \quad (3)$$

where α_i is the salience of the *i*-th stimulus, β is a learning rate, γ is the discount factor and X_0 represents the absence or presence of the reinforcer. Equation (1) gives the associative value of component *j*-th of stimulus *i*-th on the next time-step. This is based on the current value added to the temporal difference error (an estimate of how wrong the previous prediction was, based on existing information), modulated by $\bar{X}_{i,j}(t)$, a trace indicating the extent to which it is eligible for modification. The prediction, $P(t)$ given by the second equation is defined as the sum of the associative values of all the components present at that time, across all stimuli. We have developed a simulator that computes CSC TD according to the algorithm in Table 1.

TABLE 1 ABOUT HERE

Table 1. CSC TD algorithm as computed in the TD Simulator.

The TD Simulator implements a wide range of learning procedures. In particular, it runs forward, backward and simultaneous conditioning with stimuli and inter-trial intervals of fixed and variable lengths. It also works with a variety of contexts, compound stimulus and context-stimulus compounds, and with configural cues as well. The user can also modify the US parameters from phase to phase, choose between different eligibility traces, and work with any time-step size. The TD Simulator generates numerical and graphical outputs and permits the user to export the results to a data processor spreadsheet for further manipulation and analysis of data. The graphical interface allows procedures to be entered in a way that resembles standard associative learning designs.

Information Sharing Statement

The TD Simulator software is publicly and freely available from the CAL software resource page (http://www.cal-r.org/index.php?id=TD-sim), which is developed and maintained at the Centre for Computational and Animal Learning Research Ltd. All software, information and support are provided online at the TD Simulator webpage. The TD Simulator will run on any platform provided that the Java Runtime Environment (JRE) is installed. Mac and most Linux distributions already include JRE. Additionally, the user can download executable versions for Apple and Windows. The simulator is cross-platform, does not require any special equipment, operating system or support program, and does not need installation.