



LUND UNIVERSITY

Cognitive modeling with context sensitive reinforcement learning

Balkenius, Christian; Winberg, Stefan

Published in:

Proceedings of AILS 04 (Report / Lund Institute of Technology, Lund University ; 151)

2004

[Link to publication](#)

Citation for published version (APA):

Balkenius, C., & Winberg, S. (2004). Cognitive modeling with context sensitive reinforcement learning. In J. Malek (Ed.), *Proceedings of AILS 04 (Report / Lund Institute of Technology, Lund University ; 151)* (pp. 10-19). Department of Computer Science, Lund University.
<http://www.lu.se/People/Christian.Balkenius/PDF/Balkenius.Winberg.2004.pdf>

Total number of authors:

2

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Cognitive Modeling with Context Sensitive Reinforcement Learning

Christian Balkenius Stefan Winberg

Lund University Cognitive Science
Kungshuset, Lundagård
S-222 22 Lund, Sweden

1 Introduction

A reinforcement learning system is typically described as a black box which receives two types of input, the current state, S , and the current reinforcement, R . From these two inputs, the system has to figure out a policy that determines what action to perform in each state to maximize the received reinforcement in the future (Sutton & Barto, 1998). The future expected reinforcement can be estimated either by using the sum of all future reinforcement or with an exponentially decaying time horizon. It is also possible to only take into account the reinforcement received at the next goal action which results in finite horizon algorithms (e. g. Balkenius & Morén, 1999). Learning is viewed as the formation of associations between states and actions and are represented by numerical values that are changed during learning.

In most basic reinforcement learning algorithm, the policy for each state is learned individually without regard for the similarity between different states. It would obviously be valuable if actions learned in one state could be generalized to other similar states. Such generalization can be introduced into a reinforcement learning algorithm in several ways. One possibility is to code the similarity between states by similar state vectors. Such methods have been proposed by Sutton (1996), who used a tile representation or the underlying state space and Balkenius (1996), who used a multi-resolution representation. As alternative is to learn the underlying state representation during exploration based on the closeness of different states (Dayan, 1993). In both cases, learning becomes faster since each learning instance will be generalized to many similar states.

In many cases, it makes sense to divide the state input into two parts, one that code for the situation or context and one that codes for the part of the state that controls the action (cf. Balkenius & Hulth, 1999, Houghes & Drogoul, 2001). If such a combined representation is used together with the reinforcement algorithms described above, learning will generalize not only to similar states but also to similar contexts. The role of state and context will thus be symmetric.

In a series of experiment with rats, Bouton (1991) has showed that the way animals generalize learned behavior has one additional, and very important, feature. Although a learned response is generalized to similar stimuli in much the way that this is handled by reinforcement learning algorithms, learning when the generalization fails is very different and depends on the context.

In Bouton’s experiment, and others like it, rats were trained to produce a response R_1 when stimulus S_1 was presented in context A . The animals were subsequently moved to a new context B and tested there with stimulus S_1 . The learning in A generalized to B and they performed the same response R_1 in the second context. Now, the animals were asked to refrain from response R_1 in B using an extinction procedure. When they had learned not to produce R_1 in B they were moved back to the initial context A . They now performed the initially learned response R_1 again. The animals were also tested in a third context C in which they also produced the initially learned response R_1 when S_1 was presented.

This suggests that animals adhere to a generalization strategy where any learned association is first maximally generalized to new contexts and later made more specific by excluding contexts where the learned responses fail. Here, we want to show how a popular reinforcement learning algorithm, Q-learning (Watkins & Dayan, 1992), can be adapted to this framework by incorporating a second set of inputs that code for the context together with a new update rule that takes the contextual inputs into account.

In the algorithm, the two inputs for state and context are handled in an asymmetric way where the state vector is handled as usual, but the context input only influences the output through its modulation of the learned associations. In Balkenius & Morén (2000) and Morén (2002) it was shown how the context could modulate learning in a classical conditioning situation, that is, a learning task where the value of state has to be estimated without regard of the possible actions. Below, we investigate how the context can influence reinforcement learning and test it in a number of computer simulations.

2 Contextual Reinforcement Learning

To develop a context sensitive reinforcement learning algorithm, we will start from ordinary Q-learning (Watkins & Dayan, 1992). Let s_t be the current state and a_t the selected action at s_t . The result of performing action a_t in state s_t is s_{t+1} . The algorithm attempts to estimate a function $Q(s, a)$ which can be seen as the associative strength between state s and action a . The Q-function is updated as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Delta Q_t,$$

where,

$$\Delta Q_t = \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right].$$

In the simplest implementation of the algorithm, s and a are discrete and $Q(s, a)$ is represented by a table with entries $Q(s, a) = q_{s,a}$ that are changed by the update rule. The value α is the learning rate and γ is the temporal discount factor (See Sutton & Barto, 1998)

To allow better generalization or to have a more compact representation of the Q-function, different forms of function approximators can be used instead of a table. For example, it is common to use an artificial neural network, such as the backpropagation network to approximate $Q(s, a)$. Here we will use a simple linear approximator to show the general ideas.

Let each state be represented by a state vector $s = \langle s_0, s_1, \dots, s_n \rangle$ and let $\{a_0, a_1, \dots, a_m\}$ be a discrete set of actions. The Q-function is estimated as,

$$Q(s, a_j) = \sum_{i=0}^n s_i w_{ij},$$

and the update rule translates into

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \alpha s_i a_j \Delta Q_t.$$

where $a_j = 1$ for the selected action j . That is, each weight is updated according to the error in the Q-function multiplied with the value of the state component s_i . This means that only components of the state that contributed to the selected action will be updated.

It is clear that the linear approximator will generalize learning to states that are similar to each other. We will not dwell on the properties of this simple formulation here however, but instead go on to see how the context can be used in the learning rule. Let the context be described by a vector $c = \langle c_0, c_1, \dots, c_p \rangle$. According to the experiment by Bouton (1991) described above, initial learning should only depend on the state (or stimulus) while relearning when expected reinforcement is not received should depend on the context (and presumably also the state). Taking this into account, we can reformulate the linear estimator in the following way by including additional weights u_{ijk} which relates each association w_{ij} to the context c_k :

$$Q(c, s, a_j) = \sum_{i=0}^n s_i w_{ij} I_{ij},$$

where,

$$I_{ij} = \prod_{k=0}^p (1 - c_k u_{ijk}).$$

In neural network terms, I_{ij} can be seen as shunting inhibition from the context of the association from the state to the action (Fig. ??). We now need to consider how the learning rule should be changed to reflect the new context sensitive estimator.

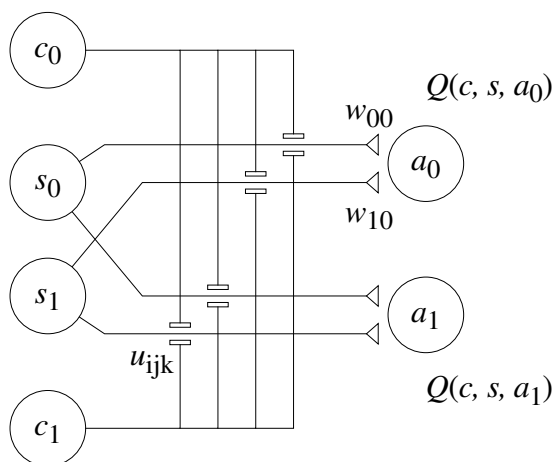


FIGURE 1: The approximation of $Q(c, s, a_j)$ as an artificial neural network with shunting inhibition from the context nodes c_k to the association between a state node s_i and an action node a_j .

When all $u_{ijk} = 0$, the algorithm work exactly as before which implies that the original equation can still be used for the case when $\Delta Q_t > 0$. This will result in initial learning that is totally independent of the context. On the other hand, when $\Delta Q_t < 0$, instead of changing the weights w_{ij} , we increase the inhibition from the current context according to

$$u_{ijk}^{(t+1)} = u_{ijk}^{(t)} - \beta s_i a_j c_k \Delta Q_t.$$

In other words, the inhibition from the current context will increase to the association between the current state and the selected action when the actual reinforcement is lower that the expected reinforcement. This captures the basic intuition of Bouton's experiment, but has one problem. Once the weights w_{ij} has reached their maximal values, all learning will take place in u_{ijk} . Also, if the appropriate action within a fixed context changes, it may become necessary to decrease the values of u_{ijk} . The solution to these problems is to allow changes in both directions of both w_{ij} and u_{ijk} , but to modulate it with the sign of ΔQ_t .

The simplest scheme is to use two learning rate constants α^+ and β^+ , which are used when $\Delta Q_t > 0$, and two constants α^- and β^- , which are used when $\Delta Q_t < 0$, and to update both w_{ij} and u_{ijk} at each time step. This results in the reinforcement learning algorithm summarized in box 1.

```

Initialize  $w_{ij} = 0$  and  $u_{ijk} = 0$ 
Repeat for each epoch
. Initialize  $s$ 
. Repeat
. . Choose  $a$  from  $s$  using policy derived from  $Q$ 
. . Take action  $a$ , observe  $r$  and  $s'$ 
. . Calculate  $\Delta Q_t$ 
. . if  $\Delta Q_t > 0$  then
. . .  $w_{ij}^{(t+1)} = w_{ij}^{(t)} + \alpha^+ \Delta Q_t$ 
. . .  $u_{ijk}^{(t+1)} = u_{ijk}^{(t)} - \beta^+ s_i a_j c_k \Delta Q_t$ 
. . else
. . .  $w_{ij}^{(t+1)} = w_{ij}^{(t)} + \alpha^- \Delta Q_t$ 
. . .  $u_{ijk}^{(t+1)} = u_{ijk}^{(t)} - \beta^- s_i a_j c_k \Delta Q_t$ 
. until  $s$  is the goal state

```

3 Experiments

The contextual reinforcement learning algorithm was tested on a number of cognitive learning problems to investigate its ability to include context in learning. First we tested the algorithm on a version of Bouton's experiment. A cognitively more interesting version of this experiment is called task switching and it was investigated next. We also tried the algorithm on a the Wisconsin Card Sorting Test which is standard diagnostic test for frontal brain injury. Finally, the algorithm was tested on a classical context sensitive categorization task.

3.1 Contextual Control

We tested the algorithm in a simple classical conditioning test where only a single state (or stimulus) s and a single action (or response) a was available. The learning rates where $\alpha^+ = 0.2$, $\alpha^- = 0$, $\beta^+ = 0$, and $\beta^- = 0.2$, that is, learning was maximally asymmetric. Since classical conditioning was tested, the reinforcement followed the *stimulus* rather than the response as is the case for reinforcement learning proper.

This response was first acquired with the context vector $c^A = \langle 1, 0, 0 \rangle$ by following any presentation of the stimulus with reinforcement 20 times. $Q(c, s, a)$ was here treated as the probability of responding.

When the response was consistently produced when the stimulus was present, the context vector was changed to $c^B = \langle 0, 1, 0 \rangle$. As expected, the response did not change by this manipulation. Now the reinforcement was withheld, and the response extinguished during 20 trials, that is, $Q(c^B, s, a)$ approached 0. We also tested the system in the original context c^A where the response now reappeared.

Finally, we tested the system with a novel context vector $c^C = \langle 0, 0, 1 \rangle$. Again the response consistently followed the presentation of the stimulus. The result shows that learning by the algorithm parallel the generalization of animals by

first generalizing learned behavior to all contexts and by only making learning context specific after relearning.

3.2 Task Switching

In a task switching experiment, the participants have to learn to respond to the same stimuli in two different ways depending to the current context or instruction. For example, in an experiment reported by Cepeda, Cepeda, & Kramer (2000), subjects were shown one of four stimuli on a screen: 1, 3, 111 or 333. Two responses were available: 1 and 3, and the context was either the question “How many?” or the question “What digit?” It is clear that the correct responses to the four stimuli changes when the context changes.

We tested the ability of the algorithm to learn this context sensitive mapping. The learning rates were again $\alpha^+ = 0.2$, $\alpha^- = 0$, $\beta^+ = 0$, and $\beta^- = 0.2$. The first question was represented by the context $c^{How\ many?} = \langle 1, 0 \rangle$ and the second by the context $c^{What\ digit?} = \langle 0, 1 \rangle$. The four stimuli were coded by a four component stimulus vector, with each component representing one of the stimuli. The system was presented with stimuli and contexts in random order and was allowed to try out the different responses according to its current policy. After 29 trials, the system reached a success rate of 100%. This shows that the algorithm can learn two different stimulus–response mappings that can be shifted as the task (or context) demands.

In Balkenius & Björne (2001), we hypothesized that the problems children diagnosed with ADHD have in this task is due to their inability to switch and maintain the context when the task changes. The simulations reported here extends our previous result by showing how the task could be learned by the context sensitive reinforcement learning algorithm.

3.3 The Wisconsin Card Sorting Test

The Wisconsin card sorting task (WCST) is a test developed in order to evaluate the test subject’s ability to shift cognitive strategies in response to changing environmental conditions. It is widely used to investigate deficits in executive functions. In some respects it bears a resemblance to the task-switching paradigm described above.

The WCST requires the test subject to sort two identical card decks of a total of 120 cards into four stacks on the basis of the number, shape and color of geometric objects printed on the cards. Feedback is provided by the examiner after each match to allow the subject to figure out the correct classification rule. When the subject has successfully sorted ten cards, the experimenter switches the sorting rule without warning. Normal test subjects, above the age of twelve, easily learn to switch to a new rule six times during the course of the experiment while subjects with lesions in the prefrontal cortex are impaired at the task.

In the test of the algorithm on the WCST, the stimulus vector contained twelve components that represented the features of the shown card. Color, number and shape were represented by four features each. There were three contexts

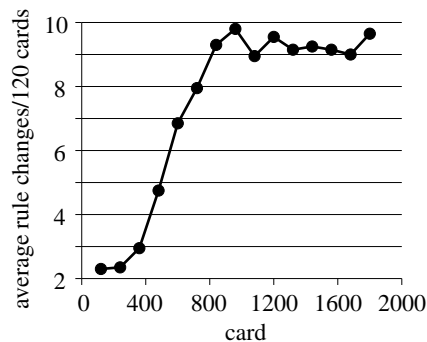


FIGURE 2: Average rule changes per 120 cards in the Wisconsin Card Sorting Test. Average of 30 runs.

coding for the sorting rule. shape, color or number, and four actions corresponding to which of the four stacks where the card should be placed. The learning rate was set to 0.1 for α^+ and β^- and to 0 for the other constants. The context was selected at random and if the current sort is incorrect, another context would be used for the next card.

The average rule changes per 120 cards was recorded as a measure of performance (Fig. ??). After five training periods, or 600 cards, the average performance of the algorithm is comparable to that of an average person.

3.4 Categorization in Context

In our final experiment we reproduced the context effects in a classical categorization experiment (Labov, 1973). In this experiment, subjects were presented pictures of containers as shown in Fig. ?? and were asked to categorize the objects. In some instances, either flowers or potatoes were placed in the containers. When flowers were present, the object was more likely to be categorised as a bowl than when the object was presented on its own, and when it contained flowers, it was more likely to be categorized as a vase.

To see whether the algorithm would reproduce this result we coded the shape of the containers in a single dimension roughly corresponding to height/width. The shape along this dimension was subsequently coded in a distributed fashion in 46 stimulus nodes using a gaussian activation covering five nodes around the node for the current object shape. The learning rates were set to $\alpha^+ = 0.1$ and $\beta^- = 0.2$ and 0 for the other constants. There were three contexts coding for ‘flowers’, ‘potatoes’ and ‘nothing’ respectively, and two outputs corresponding to the categories ‘bowl’ and ‘vase’.

A subset of 66 example stimuli were selected for the training and the system was tested on the full set of 126 examples after three epochs. If the experiment was run for longer than three epochs, overfitting would occur. The results are shown in Fig. ?. The decision border between the two categories is shifted as

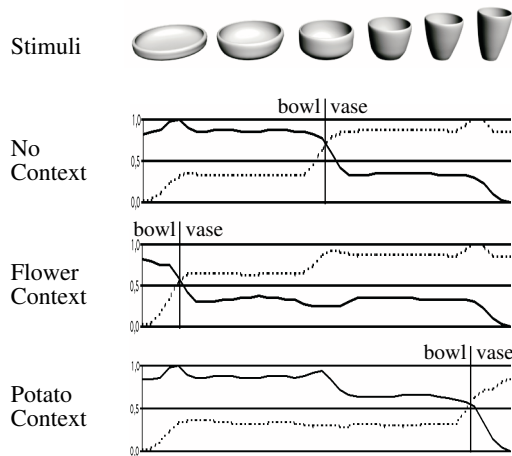


FIGURE 3: *Generalization curves for the two categories ‘vase’ and ‘bowl’ as functions of the shape of the stimulus and the context. The decision border between the categories moves when the context changes.*

expected when the context indicates that the object is bowl or a vase.

4 Discussion

We have described how a standard reinforcement learning algorithm can be changed to include a second contextual input that is used to modulate the learning in the original algorithm. The new algorithm takes the context into account during relearning when the previously learned actions are no longer valid. The algorithm was tested on a number of cognitive experiment and shown to reproduce the learning in both a task switching test and in the Wisconsin Card Sorting Test. In addition, the algorithm was able to learn a context sensitive categorization of objects in the Labov experiment.

There are several areas where the model could be investigated further. One important area for further study is the relation between stimulus and context generalization that has only very briefly been touched upon here. How does the context influence the generalization of an action to similar states and how does the learning history influence this?

It is also interesting to relate the concept of a context to that of a goal. When a sequential policy has been learned, a goal representation as part of the context can be used to select actions that are consistent with that goal. On a smaller scale, part of the context could be used to code preconditions for an action that would be automatically exploited by the algorithm to exclude actions where the precondition is not met.

As a cognitive model, context is closely related to working memory, a rela-

tion that will be further investigated in the future. We will also compare the new algorithm with other learning methods to evaluate to what extent the asymmetric use of state and context enhances learning. The learning method will also be formally analyzed.

Acknowledgements

The code used for the simulations described in this article are available as part of the Ikaros project at <http://www.lucs.lu.se/Ikaros>.

References

- Balkenius, C. (1996). Generalization in instrumental learning. In Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S. W. (Eds.), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press/Bradford Books.
- Balkenius, C., and Björne, P. (2001). Toward a robot model of attention-deficit hyperactivity disorder (ADHD). In Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., and Breazeal, C. (Eds.), *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. Lund University Cognitive Studies, 85.
- Balkenius, C., and Morén, J. (1999). Dynamics of a classical conditioning model. *Autonomous Robots*, 7, 41-56.
- Balkenius, C. and Hulth, N. (1999). Attention as selection-for-action: a scheme for active perception. In Schweitzer, G., Burgard, W., Nehmzow, U., and Vestli, S. J. (Eds.), *Proceedings of EUROBOT '99* (pp. 113–119). IEEE Press.
- Balkenius, C., & Morén, J. (2000). A computational model of context processing, In J.-A. Meyer, A. Berthoz, D. Floreano, H. L. Roitblat, S. W. Wilson, (Eds.) *From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behaviour*, (pp. 256–265), Cambridge, MA: The MIT Press.
- Bouton, M. E. (1991). Context and retrieval in extinction and in other examples of interference in simple associative learning. In L. W. Dachowski and C. F. Flaherty (Eds.), *Current topics in animal learning: Brain, emotion, and cognition* (pp. 25–53). Hillsdale, NJ: Erlbaum.
- Cepeda, N.J., Cepeda, M.L., Kramer, A.F. (2000). Task switching and attention deficit hyperactivity disorder, *Journal of Abnormal Child Psychology*, 28, 3, 213-226.
- Dayan, P. (1993) Improving generalization for temporal difference learning: the

- successor representation. *Neural Computation*, 5, 613–624.
- Morén, J. (2002). *Emotion and Learning: A Computational Model of the Amygdala*, Lund University Cognitive Studies, 93.
- Labov, W. (1973). The boundaries of words and their meanings, In *New Ways of Analyzing Variation in English*, J. Fishman, (Ed.), Georgetown University Press, Washington, DC, 340–373.
- Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coding. In D. S. Touretzky, M. C. Mozer and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems: Proceedings of the 1995 Conference* (pp. 1038–1044). Cambridge, MA: MIT Press.
- Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.
- Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning. *Machine Learning*, 9, 279–292.
- Houghes, L., and Drogoul, A. (2001). Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., and Breazeal, C. (Eds.). *Proceedings of the first international workshop on epigenetic robotics: modeling cognitive development in robotic systems*. Lund University Cognitive Studies, 85.