Categorization and Reinforcement Learning:
State Identification in Reinforcement Learning and Network Reinforcement Learning

Vladislav D. Veksler (vekslv@rpi.edu)
Wayne D. Gray (grayw@rpi.edu)
Michael J. Schoelles (schoem@rpi.edu)
Cognitive Science Department, 110 8th Street
Troy, NY 12180 USA

Abstract

We present Network Reinforcement Learning (NRL) as more efficient and robust than traditional reinforcement learning in complex environments. Combined with Configural Memory (Pearce, 1994), NRL can generalize from its experiences to novel stimuli, and learn how to deal with anomalies as well. We show how configural memory with NRL accounts for human and monkey data on a classic categorization paradigm. Finally, we argue for why the suggested mechanism is better than other reinforcement learning and categorization models for cognitive agents and AI.

Keywords: categorization, reinforcement learning, category learning, unsupervised learning, cognitive modeling, cognitive architectures, artificial intelligence, configural.

Introduction

A red line is not just red, nor is it just a line, nor is it just a red line. It is all of these things at the same time. Given a red line, an agent may want to select red-appropriate actions, line-appropriate actions, or red-line appropriate actions. Identifying the object as a red-line may be inefficient, and identifying it as red may be misleading.

In this paper we identify problems with state identification in Reinforcement Learning and suggest a mechanism that addresses these problems, Configural Memory with Network Reinforcement Learning (CMNRL). We will argue that CMNRL is more efficient and robust than either instance-based or category-based reinforcement learning. We will then describe a classic categorization task (Shepard, Hovland, & Jenkins, 1961), and suggest the psychological validity of CMNRL by simulating human and monkey data from this task.

Problems with Reinforcement Learning

It has become popular to use some form of reinforcement learning in computational cognitive agents in order to get computational agents to act in a psychologically and biologically plausible manner (e.g. Fu & Anderson, 2006; Holroyd & Coles, 2002; Sutton & Barto, 1998). Reinforcement learning (RL) is a procedural component of a cognitive agent that can be described as follows: the agent must identify its current state, *S*, and then execute some action, *A*, such that the state-action pair, *S–A*, has the highest utility of all state-action pairs for state *S* (Figure 1). The utility of the selected state-action pair, *U(S–A)*, is updated based on environmental feedback in the form of a reinforcement signal (e.g. pleasure/pain). Exploratory and learning mechanisms vary from one version of RL to another, but the basic idea remains.

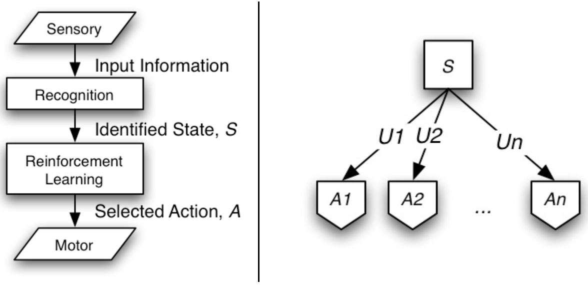

Figure 1. Reinforcement Learning. Left: Flow of information in RL agents (no learning shown). Right: Identified state, *S*, and possible actions, *A1* through *An*; each arrow represents a competing state-action pair; arrow labels *U1* through *Un* represent the utilities of state-action pairs *S–A1* through *S–An*.

The major problem with reinforcement learning is state recognition. Instance-based state identification, where each unique set of input activations is considered a different state, is largely inefficient due to the exploding number of state-action pairs with each new input channel. Instance-based RL could require 2^n states for *n* binary inputs, and $a \cdot 2^n$ state-action pairs, where *a* is the number of actions.

To decrease the decision space, researchers use various forms of categorization to preprocess large numbers of input channels into more manageable numbers of states (for review see Sutton & Barto, 1998). However, creating too many categories still results in too large of a decision/learning space. As the number of categories increases, the category-based agent becomes more similar to an instance-based one. Alternatively, bunching up objects into a small number of categories may misrepresent the environment altogether.

Consider the following simplified scenario. Imagine a world where all apples taste great except brown apples (*brown-apple* is the exception to the *apple* category). Instance-based and category-based RL agents are presented with two red, one green, and two brown apples each (Table 1). Let us assume that the category-based agent in this scenario contains the 'apple' category. At the end of the five stimulus-action-reinforcement cycles presented in Table 1, the category-based agent would know only one thing – that eating apples has a positive utility (+1), and thus would continue eating brown apples. At the end of the same five cycles, the instance-based agent would know the utility of eating red apples (+2), green apples (+1), and brown apples (-2), but would know nothing about eating apples in general. Presented with a yellow apple, the instance-based agent will not know what to do with it, and may throw it away. In this world, instance-based agents will never know what to do with newly encountered apples, whereas category-based ones will never learn to avoid rotten apples. A more efficient agent should be able to learn about both categories and instances.

Table 1. Instance-based and Category-based Reinforcement Learning. Two columns on the right display the updated utilities after each I-A-R cycle. (I = input; A = action; R = reinforcement)

I	A	R	Instance-based RL	Category-based RL
red apple	Eat	+1	U(red-apple,Eat) = +1	U(apple,Eat) = +1
green apple	Eat	+1	U(green-apple,Eat) = +1	U(apple,Eat) = +2
red apple	Eat	+1	U(red-apple,Eat) = +2	U(apple,Eat) = +3
brown apple	Eat	-1	U(brown-apple,Eat) = -1	U(apple,Eat) = +2
brown apple	Eat	-1	U(brown-apple,Eat) = -2	U(apple,Eat) = +1

Network Reinforcement Learning

We can imagine an agent with a dynamic state-space, such that the agent could identify categories and category-exceptions as needed. Such an agent may be able to use category-based state identification for all apples except brown apples in the example from Table 1.

Here is the problem: we cannot disregard the fact that a brown apple is brown, or that it is an apple. It may be the case that action *x* (e.g. *eat*) results in different reinforcement values between *brown-apple* and *apple* states (*brown-apple* is an exception to the *apple* category), and action *y* (e.g. *throw*) is the same for the *brown-apple* state as it is for its parent category (*brown-apple* follows the *apple* category rule). Moreover, even if *brown-apple* was originally hypothesized by the agent to be an exception to its parent categories, we want to allow the agent to learn through experience whether that hypothesis is true or not. If the agent fails to forget useless exceptions it will continuously shift towards the inefficient instance-based state identification.

Continuing with our example, if we do not disregard the parent categories of the identified exception, then we have a model of reinforcement learning that recognizes a brown apple as three different states – *brown*, *apple*, and *brown-apple*. Let us say that we consider all of the state-action pairs of all of these states, and choose some action, *A*, and receive some reinforcement, *r*. Does this mean that *r* belongs to *brown–A*, *apple–A*, or *brown-apple–A*?

To make this more concrete, if you eat a brown apple, and it tastes bad, is it because it was an apple, because it was brown, or because it was a brown apple? Maybe it is time for you to learn that apples just do not taste good. Maybe apples taste great but brown apples do not. Maybe all brown things taste bad. The only solution from the model's perspective is to propagate the reinforcement value to all active states, and hope that the utility of the *apple–eat* state is already high enough that the negative reinforcement does not significantly hurt it (unfortunately, if the 'bad apple' phenomena happens early enough, people do get stuck with food aversions).

Configural Memory with Network Reinforcement Learning

Using simultaneous activation of multiple states and simultaneous learning of multiple state-action pair utilities (hereafter Network Reinforcement Learning, NRL) has been previously suggested by Porta & Celaya (2005). They argued that NRL is more efficient than traditional RL for 'real world' robot learning. The basic idea behind our adaptation of NRL is as follows: unlike traditional reinforcement learning where a single state *S* is identified, in NRL we identify multiple states, *{S1,S2,...,Sn}* (Figure 2; left). Given *n* number of identified states and *m* number of possible actions, there are $n{\cdot}m$ competing state-action pairs (Figure 2, right; each arrow represents a competing state-action pair). The state-action pair with the highest utility is chosen and the winning action, *A*, is activated. Unlike traditional RL where only the winning state-action pair is updated, the utility values for all *s–A* state-action pairs are updated with the reinforcement feedback, s∈*{S1,S2,...,Sn}*.

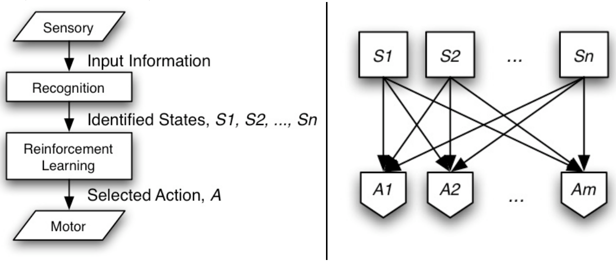

Figure 2. Network Reinforcement Learning. Left: Flow of information in NRL agents (no learning shown). Right: Identified states, *S1* through *Sn*, and possible actions, *A1* through *An*; each arrow represents a competing state-action pair.

Porta & Celaya (2005) used feature detectors (higher order inputs; e.g. vertical-line and hand-shape neurons are considered feature-detectors in humans) as competing states. Where our theory differs from Porta & Celaya is in our use of incremental configural memory instead of arbitrarily preclassified feature detectors. Configural representation is very powerful, and is psychologically validated via habituation and discrimination studies (for review see Pearce, 1994). Configural memory is more expressive than feature detectors because it may include all possible input configurations (including feature detectors). For example, for three inputs (e.g. white, square, large) there could be as many as seven configural nodes (e.g. white, square, large, white-square, white-large, large-square, large-white-square).

Configural representations are sometimes criticized for the exploding number of configurations (e.g. Heydemann, 1995): for stimuli that vary across 3 binary dimensions (e.g. small/large, black/white, triangle/square) a configural model would need 26 configural nodes (e.g. small, large, black, white, triangle, square, small-black, small-white, small-square, ..., small-white-square, small-white-triangle). Given a stimulus varying on 6 binary dimensions, the number of possible configural nodes would rise to 126, etc. This problem has been addressed in the IAK model (Heydemann, 1995), which uses probabilistic sampling to select a small subset of configurations, thus avoiding the rapid growth.

Combining configural memory with NRL (CMNRL) provides for simultaneous multi-level state identification, such that types and tokens of every level may be used as reinforcement learning states. In this type of a model there may be different actions attributable to objects at every level. For example, *animal*, *dog*, *golden retriever*, and *my golden retriever named Sparky* may all have some actions in common and some actions that set them apart. Upon seeing Sparky, all of these actions would compete. Upon action-feedback, learning would occur for all of the activated states, from top-level *dog* to bottom-level *Sparky*.

CMNRL versus RL

CMNRL is more efficient and robust than traditional Reinforcement Learning. To observe this, consider the case in Table 1. Instance-based RL would fail to generalize and make any predictions about a new apple object. Category-based RL would fail to learn about the negative utility of eating a brown apple. Due to the fact that CMNRL updates state-action pairs at both object and category levels simultaneously, it is able to generalize, and learn about exceptions, as well.

CMNRL is not a mere combination of a categorization model with reinforcement learning. Integration of RL or NRL with one of the existing categorization models, such as RULEX (Nosofsky, Palmeri, & McKinley, 1994) or SUSTAIN (Love, Medin, & Gureckis, 2004), will require twice as much feedback for learning. Such integration would produce an agent that requires supervised learning to identify declarative category structure, and then reinforcement learning to learn procedural utilities. In contrast, CMNRL considers procedural memory part of the categorization process, and thus requires only the reinforcement signal as feedback. In other words, CMNRL uses the reinforcement signal to learn about category structure.

In the following sections we will describe a classic categorization experiment, and compare CMNRL results against human and monkey data from this experimental paradigm. We will analyze how CMNRL performs this task, and contrast it against traditional reinforcement learning with and without categorization. We will also discuss how standard categorization models perform on this task, and where CMNRL stands out from these.

Shepard, Hovland, & Jenkins, 1961

The beauty of the Shepard, Hovland, & Jenkins (1961) benchmark categorization experiment is in its simplicity. Subjects are presented with one of eight objects varying across three binary dimensions (e.g. small/large, black/white, triangle/square), and have to pick one of two responses (e.g. A or B). Feedback is then provided as to whether the response was correct.

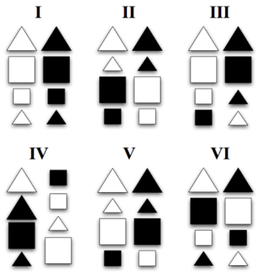

Figure 3. An example of the six types of categorization problems from the Shepard, Hovland, & Jenkins task. For each type, subjects must learn that the items in the left column are one category, whereas the items in the right column are another.

The idea behind this study is to determine the rate at which people learn to classify each of the eight objects as belonging to one of two categories. Four of the objects belonged to category A, and the other four belonged to category B. Given this setup, there are only six different types of possible category breakdowns. In the example in Figure 3 the three binary dimensions are shape (square vs. triangle), color (black vs. white), and size (large vs. small); categories A and B are represented as left and right columns; and the six problem types are marked with roman numerals I – VI.

The general results of the study indicated that human performance for problem types I through VI follow the order: I > II > (III, IV, V) > VI. These results have been replicated in multiple forms (Love, 2002; Nosofsky, Gluck, Palmeri, McKinley, & Glauthier, 1994; Smith, Minda, & Washburn, 2004), each time confirming the main effect found by Shepard, Hovland, & Jenkins over forty-five years ago. It is rather simple to explain and model the fact that people performed best on problem I and worst on problem VI – while only one dimension is necessary to predict the category (A or B) in type I problems, all three dimensions are necessary for correct category identification in type VI. What is less obvious for most models of categorization is that performance on problem II is better than that on types III, IV, and V.

Supervised and Unsupervised Category Learning in the Shepard, Hovland, & Jenkins categorization task

In further investigation as to why problem II was learned faster than the problem IV, Love (2002) hypothesized that this was a result of the learning mode – namely, supervised classification learning. In supervised classification learning the subject is presented with a stimulus, they give a response, and then corrective feedback is provided. Given that much of category learning occurs in unsupervised, and sometimes completely incidental fashion, supervised classification learning is not sufficient in explaining and predicting all human categorization behavior.

Love omitted problems III and V, and changed his experimental procedure so that supervised classification learning could be directly compared with what he called unsupervised-intentional and unsupervised-incidental learning modes. Unfortunately the details of the procedure are beyond the scope of this paper. The results from the supervised category learning condition of Love (2002) followed in order with the findings from Shepard, Hovland, & Jenkins (1961) and Nosofsky, Gluck, et al. (1994). Subject performance on problem types I through VI followed the order: I > II > IV > VI. Unsupervised category learning, however, lead to performance differences in this experiment. In particular, during these less volitional modes of category learning, performance was better on type IV than type II problems (see Figure 4 for results). Interestingly enough, further investigation of this paradigm by Smith et al. (2004), revealed that rhesus monkeys performed better on type IV problems than type II problems, as well.

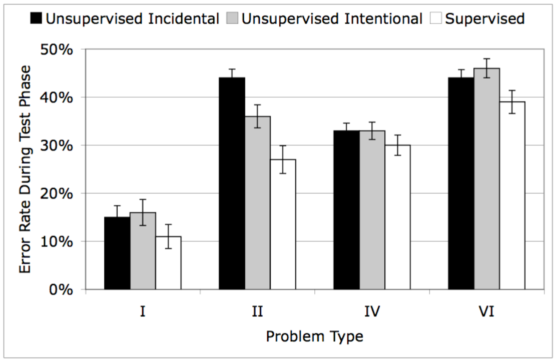

Figure 4. Subject performance during the test phase of Love (2002). Error bars represent standard error.

Modeling the Shepard, Hovland, & Jenkins paradigm

The Shepard et al. paradigm has become one of the benchmarks for models of categorization. Among the models that have tried to model this task are: configural-cue model (Gluck & Bower, 1988), DALR (Gluck, Glautheir, & Suton, 1992, as cited by Nosofsky, Gluck, et al., 1994), the rational model (Anderson, 1991, as cited by Nosofsky, Gluck, et al., 1994), ALCOVE (Kruschke, 1992), ALCOVE-RL (Phillips & Noelle, 2004), RULEX (Nosofsky, Palmeri, & McKinley, 1994), SUSTAIN (Love, Medin, & Gureckis, 2004), and IAK (Heydemann, 1995). The latter five of these eight models produced the same problem performance ordering as the human supervised classification data would suggest: I > II > (III, IV, V) > VI. However, the models that failed at replicating human supervised classification data might have actually provided good fits to unsupervised category learning data from Love (2002) and rhesus monkey data from Smith et al. (2004), predicting the advantage in performance on problems III, IV, V over problem II. None of the leading categorization models have attempted to explain both modes of category learning.

Among the failed attempts to model human supervised classification data was the configural-cue model, which very closely resembles the CMNRL setup on this task. The major difference between CMNRL and configural-cue is that CMNRL uses NRL instead of the least mean squares rule learning used by the configural-cue model. The use of NRL allows CMNRL to capture exploration and trial-and-error learning. It also affords the use of unsupervised learning, where the environment feeds back a reinforcement signal instead of suggesting the correct category for each object. This distinction is most important because supervised forms of learning are not always available in natural environments. The necessity for supervised learning is a problem with all categorization models cited above.

Modeling Shepard, Hovland, & Jenkins, 1961 with CMNRL

Traditionally the Shepard, Hovland, & Jenkins paradigm was viewed as a study of declarative category learning; that is, each of the eight objects had to be classified as category A or B. We assume, however, that since the subjects had to respond, there is a procedural component involved. Thus we use NRL to model the responses, and a configural representation to identify the stimuli.

As previously mentioned, for stimuli that vary across 3 binary dimensions, as in Figure 3, a configural model would need a maximum of 26 configural nodes. For the simple stimuli used in this experimental paradigm we assume that human subjects have all 26 possible configurations in memory. With only two possible responses (e.g. A and B), there are a total of 52 state-action pairs (Figure 5).

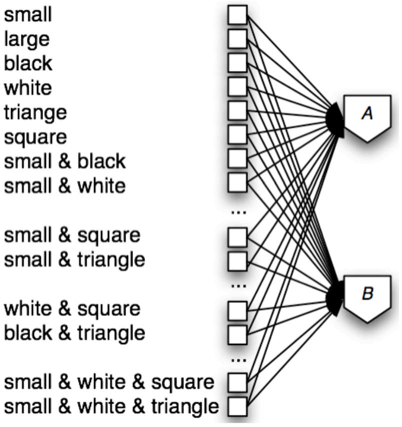

Figure 5. CMNRL setup for the Shepard, Hovland, & Jenkins (1961) paradigm.

There are three free parameters in the current implementation of NRL – utility of exploration (*Ue*), perceived utility of positive reinforcement (*U*+), and perceived utility of negative reinforcement (*U-*). Every trial the model was presented with three known dimensions and had to answer the fourth. If the exploration parameter, *Ue*, was higher than utility values of all competing state-action pairs, the model would try an action at random; otherwise the model would activate the action of the state-action pair with the highest utility value. If the model answered correctly, positive reinforcement, *U+*, would propagate to all active state-action pairs. If the model answered incorrectly, negative reinforcement, *U-*, would propagate to all active state-action pairs.

For example, if the model saw *large-white-square* in condition IV of Figure 3 there would be seven configural nodes active (*white*, *square*, *large*, *white-square*, *white-large*, *large-square*, and *large-white-square*), and fourteen competing state-action pairs. If *white-square–B* had the highest utility of all other state-action pairs, and had a higher utility than the parameter *Ue*, the model would execute action *B*. In this case, the answer would be correct, and the utilities for all seven active state-action pairs (*white–B*, *square–B*, *large–B*, *white-square–B*, *white-large–B*, *large-square–B*, and *large-white-square–B*) would be incremented by *U+*.

Given that the ratios of these parameters rather than their absolute values are of the essence, *Ue* was held constant at 1.0 for all model runs. A genetic algorithm (using the least mean square error, LMSE, criterion) was employed to find the best fitting parameters for each of the six datasets: Nosofsky, Gluck, et al. (1994), Love (2002) supervised, Love unsupervised intentional, Love unsupervised incidental, Smith et al. (2004) human, and Smith et al. rhesus monkey. Comparisons to the data from Nosofsky, Gluck, et al. (1994) study were limited to the first 192 trials only (same number of trials as Smith et al., 2004).

This is a simple setup that should demonstrate the flexibility of NRL to learn categories, and do so in a humanlike (or monkeylike) fashion. The ability of NRL to explain the differences between various category-learning modes with mere adjustments of perceived positive and negative reinforcement gives additional power to the proposed model – including more free parameters in a model could explain much more data, but would be much less meaningful; see (Su, Myung, & Pitt, 2005).

Modeling Results

The *U+,U-* parameter pairs that produced the lowest root mean square errors (RMSE) are shown in Table 2. RMSE was a stricter criterion for the model than r^2. Although higher r^2 values were found (e.g. best correlation value for Smith et al. monkey data was 0.986, RMSE=0.198), maximizing r^2 values, as opposed to minimizing RMSE values, sometimes resulted in accidental correlations (e.g. although the best correlation value for Love unsupervised incidental data was 1.000, RMSE=0.183, the actual error-rate averages on problem types I, II, IV, and VI were 47%, 50%, 49%, and 50%, respectively, signifying random behavior).

Table 2. Best parameter fits for each of the six datasets.

Fit	*U+*	*U-*	R^2	RMSE
Nosofsky, Gluck, et al. (1994)	0.22	-1.85	0.986	0.017
Love (2002) Supervised	0.65	-0.60	0.971	0.067
Love (2002) Unsupervised Intentional	6.98	-6.60	0.983	0.024
Love (2002) Unsupervised Incidental	4.56	-4.30	0.986	0.023
Smith et al. (2004) Human	0.14	-0.43	0.959	0.024
Smith et al. (2004) Monkey	0.01	-1.03	0.911	0.041

The better (lower RMSE) parameter values seemed to be related to the ratio of positive to negative reinforcement, *U+:U-*, and the average absolute reinforcement value, *U+-* (*U+-* is equivalent to the ratio of average absolute reinforcement value to the utility of exploration, *U+-:Ue*, because *Ue*=1.0). The best RMSE values for Love's experiments, each involving 80 learning trials, and 24 test trials where no learning occurred, had the *U+:U-* ratio of ≈1.1:1. For the 192-trial experiments of Nosofsky et al. and Smith et al. the best *U+:U-* ratios were ≈1:5. The top *U+:U-* ratios for the 2000-trial monkey experiment were in the range between ≈1:10 and ≈1:1000. Seemingly, in this sort of an experiment, as the number of trials increases, the average perceived positive reinforcement value drops, while the average perceived negative reinforcement value grows. Although this makes sense intuitively (when Michael Jordan misses a foul shot, he gets down on himself quite a bit more than when a novice does the same), the NRL mechanism currently in place in the CMNRL architecture does not yet account for this phenomenon.

Top *U+-* values seemed to correlate with learning mode. Of the three human supervised classification experiments – Nosofsky et al., Love supervised, and Smith et al. human – average *U+-* of the top parameter sets was 0.64. Of the unsupervised categorization experiments by Love, average *U+-* of the top parameter sets was 5.61. What this really means is that *Ue* was relatively small for supervised classification, and relatively large for unsupervised learning modes. This too makes intuitive sense – the average utility of exploration should be higher when we are learning actively, i.e. trial-and-error, as in the three supervised classification experiments.

Like ALCOVE, ALCOVE-RL, RULEX, SUSTAIN, and IAK, CMNRL was able to capture the general trend in performance across the six problem types in human supervised classification learning, going beyond other categorization models like configural-cue, DALR, and the rational model. With mere adjustments of positive and negative reinforcement values, CMNRL was also able to explain category learning in rhesus monkeys, as well as unsupervised category learning in humans.

Summary

In this paper we argued that CMNRL is more efficient and robust than either instance-based or category-based reinforcement learning. We also suggest the psychological validity of CMNRL by simulating human and monkey data from a classic categorization paradigm (Shepard, Hovland, & Jenkins, 1961).

CMNRL uses configural memory, where multiple types and tokens are activated simultaneously upon object recognition (e.g. my red robin activates {my red robin, red robins, red, robins, birds, animate objects, ...} nodes). Network Reinforcement Learning extends traditional RL to handle simultaneous activation of multiple types and tokens. Whereas traditional RL perceives the world as a single state, NRL uses multiple nodes to represent the state of the world. Whereas traditional RL reinforces one state-action pair at a time, NRL updates the utilities of all relevant state-action pairs. In this manner, CMNRL does not fail to generalize from its experiences, nor to account for anomalies.

In simulating data from the Shepard, Hovland, & Jenkins paradigm, CMNRL went beyond other categorization models. First, CMNRL is an example that it is not necessary to include a separate model of categorization along with a procedural mechanism – the procedural mechanism in CMNRL, NRL, is an essential part of the categorization mechanism. Second, CMNRL does not rely on supervised learning. This is important because the answers are not always available in the environment, only the reinforcement signals. Last, but not least, CMNRL captures and explains the data from both supervised and unsupervised modes of category learning. The differences between the two types of experiments are explained in CMNRL using the relative utility of exploration. During the supervised learning mode, subjects were more likely to explore (learn by trial and error). During unsupervised learning modes, subjects were more likely to learn passively (no exploration).

CMNRL combines declarative structure and procedural learning to make for a more complete procedural mechanism and a more complete categorization mechanism. This issue of integration makes CMNRL a better candidate of memory implementation for cognitive architectures. Without much redesign or modeling efforts, CMNRL may be used to create a cognitive agent that can learn a new environment from scratch, explore, and improve performance on arbitrary tasks. Future work on CMNRL will involve testing the mechanism in foraging environments, simple games, and language learning.

Acknowledgments

Many thanks to Ron Sun for his advice throughout this research. This work was supported, in part, by the Disruptive Technology Office, A-SpaceX contract N61339-06-C-0139 issued by PEO STRI. The views and conclusions are those of the authors, not of the US Government or its agencies.

References

Fu, W. T., & Anderson, J. R. (2006). From recurrent choice to skill learning: A reinforcement-learning model. *Journal of Experimental Psychology-General, 135*(2), 184-206.

Gluck, M. A., & Bower, G. H. (1988). A Configural-Cue Network Model of Classification Learning. *Bulletin of the Psychonomic Society, 26*(6), 500-500.

Heydemann, M. (1995). *A connectionist model for classification learning – the IAK model.* Paper presented at the Seventeenth Annual Conference of the Cognitive Science Society.

Holroyd, C. B., & Coles, M. G. H. (2002). The neural basis. of human error processing: Reinforcement learning, dopamine, and the error-related negativity. *Psychological Review, 109*(4), 679-709.

Kruschke, J. K. (1992). ALCOVE: an exemplar-based connectionist model of category learning. *Psychol Rev, 99*(1), 22-44.

Love, B. C. (2002). Comparing supervised and unsupervised category learning. *Psychon Bull Rev, 9*(4), 829-835.

Love, B. C., Medin, D. L., & Gureckis, T. M. (2004). SUSTAIN: a network model of category learning. *Psychol Rev, 111*(2), 309-332.

Nosofsky, R. M., Gluck, M. A., Palmeri, T. J., McKinley, S. C., & Glauthier, P. (1994). Comparing models of rule-based classification learning: a replication and extension of Shepard, Hovland, and Jenkins (1961). *Mem Cognit, 22*(3), 352-369.

Nosofsky, R. M., Palmeri, T. J., & McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychol Rev, 101*(1), 53-79.

Pearce, J. M. (1994). Similarity and discrimination: a selective review and a connectionist model. *Psychol Rev, 101*(4), 587-607.

Phillips, J. L., & Noelle, D. C. (2004). *Reinforcement learning of dimensional attention for categorization.* Paper presented at the Twenty-Sixth Annual Conference of the Cognitive Science Society.

Porta, J. M., & Celaya, E. (2005). Reinforcement learning for agents with many sensors and actuators acting in categorizable environments. *Journal of Artificial Intelligence Research, 23*, 79-122.

Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs: General and Applied, 75*, 1-42.

Smith, J. D., Minda, J. P., & Washburn, D. A. (2004). Category learning in rhesus monkeys: a study of the Shepard, Hovland, and Jenkins (1961) tasks. *J Exp Psychol Gen, 133*(3), 398-414.

Su, Y., Myung, I. J., & Pitt, M. A. (2005). Minimum description length and cognitive modeling. In P. D. Grünwald, I. J. Myung & M. A. Pitt (Eds.), *Advances in minimum description length: Theory and applications* (pp. 411–433). Cambridge, MA: The MIT Press.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press.