Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 12 (2011) 144–153

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

Learning to use episodic memory

Action editor: Andrew Howes

Nicholas A. Gorski *, John E. Laird

Computer Science & Engineering, University of Michigan, 2260 Hayward St., Ann Arbor, MI 48109-2121, USA

Received 22 December 2009; accepted 29 June 2010
Available online 8 August 2010

Abstract

This paper brings together work in modeling episodic memory and reinforcement learning (RL). We demonstrate that is possible to learn to use episodic memory retrievals while simultaneously learning to act in an external environment. In a series of three experiments, we investigate using RL to learn what to retrieve from episodic memory and when to retrieve it, how to use temporal episodic memory retrievals, and how to build cues that are the conjunctions of multiple features. In these experiments, our empirical results demonstrate that it is computationally feasible to learn to use episodic memory; furthermore, learning to use internal episodic memory accomplishes tasks that reinforcement learning alone cannot. These experiments also expose some important interactions that arise between reinforcement learning and episodic memory. In a fourth experiment, we demonstrate that an agent endowed with a simple bit memory cannot learn to use it effectively. This indicates that mechanistic characteristics of episodic memory may be essential to learning to use it, and that these characteristics are not shared by simpler memory mechanisms.

Keywords: Reinforcement learning; Episodic memory; Cognitive architecture; Artificial intelligence; Intelligent agents

1. Introduction

In this paper, we study a mechanism for learning to use the retrieval of knowledge from episodic memory. This unifies two important related areas of research in cognitive modeling. First, it extends prior work on the use of declarative memories in cognitive architecture where knowledge is accessed from declarative memories via deliberate and fixed cued retrievals (Anderson, 2007; Nuxoll & Laird, 2007; Wang & Laird, 2006) by exploring mechanisms for learning to use both simple and conjunctive cues. Second, it extends work on using reinforcement learning (RL) (Sutton & Barto, 1998) to learn not just control knowledge for external actions, but also to learn to control access to internal memories, expanding the range of behaviors that can learned by RL.

Earlier work has investigated increasing the space of problems applicable to RL algorithms by including internal memory mechanisms that can be deliberately controlled: Littman (1994) and Peshkin, Meulaeu, and Kaelbling (1999) developed RL agents that learned to toggle internal memory bits; Pearson, Gorski, Lewis, and Laird (2007) showed that an RL agent could learn to use a simple symbolic long-term memory; and Zilli and Hasselmo (2008) developed a system that learned to use both an internal short-term memory and an internal spatial episodic memory, which could store and retrieve symbols corresponding to locations in the environment. All four cases demonstrated a functional advantage from learning to use memory.

Our work significantly extends these previous studies in three ways: first, our episodic memory system automatically captures all aspects of experience; second, our system learns not only when to access episodic memory, but also learns to construct conjunctive cues and when to use them; and third, it takes advantage of the temporal structure of

[*] Corresponding author. Tel.: +1 734 763 0150; fax: +1 734 763 1260.
E-mail addresses: ngorski@umich.edu (N.A. Gorski), laird@umich.edu (J.E. Laird).

1389-0417/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cogsys.2010.08.001

episodic memory by learning to advance through episodic memory when it is useful (this property is also shared by the Zilli & Hasselmo system, but for simpler task and episodic memory representations).

Our studies are pursued within a specific cognitive architecture, namely Soar (Laird, 2008), which incorporates all of the required components: perceptual and motor systems for interacting with external environments, an internal short-term memory, a long-term episodic memory, an RL mechanism, and a decision procedure that selects both internal and external actions. In comparison, ACT-R (Anderson, 2007) has many similar components but does not have an explicit episodic memory. Its long-term declarative memory stores only individual chunks, and it does not store episodes that include the complete current state of the system. To do so would require storing the contents of all ACT-R's buffers as a unitary structure, as well as the ability to retrieve and access them, without having the retrieved values being confused with the current values of those buffers. Moreover, ACT-R's declarative memory does not inherently encode the temporal structure of episodic memory, where temporally consecutive memories can be recalled (Tulving, 1983). While the work presented in this paper is specific to learning to use an episodic memory, similar work could be pursued in the context of ACT-R by learning to use its declarative memory mechanism. However, we are unaware of existing work in that area, and even if there were, it would fail to engage the same issues that arise with episodic memory.

2. Background

Soar includes an episodic memory that maintains a complete history of experience (Nuxoll & Laird, 2007), implemented so as to support efficient memory storage and retrieval (Derbinsky & Laird, 2009). Soar's working memory is a relational graph structure, consisting of nodes and links, similar to the structure of a semantic network. Complete "snapshots" of working memory are automatically stored in episodic memory following every processing cycle.

To retrieve an episode, a *cue* is created in working memory by the application of Soar's procedural knowledge, which is encoded as production rules (Laird, 2008). A cue is a partial specification of an episode, created in a special part of working memory. The episode that best matches the cue is retrieved to working memory. The degree of match is based on the number of elements in the cue found in an episode. If there are multiple episodes with the same degree of match, the most recent of those episodes is retrieved. Once an episode is retrieved to working memory, other knowledge (such as procedural knowledge) can access it. This style of cue-based retrieval process is similar to ACT-R's declarative memory retrieval process where procedural knowledge creates a cue in a retrieval buffer, and the declarative memory mechanism retrieves the appropriate chunk from the long-term store.

We refer to this type of cue-based retrieval as *deliberate*, to contrast it with *spontaneous* or automatic retrieval processes. A spontaneous retrieval process is automatic and depends on all the structures in working memory. Thus, an agent with spontaneous retrieval lacks control over when retrievals take place and what aspects of the situation are the basis for retrieval, whereas with deliberate control, the agent can control when episodic memory retrievals are initiated and what cues are the basis for retrieval.

After performing a cue-based retrieval, the agent can utilize the temporal structure of episodic memory and retrieve the next episode, providing a mechanism for the agent to move forward through its memories. This allows an agent to recall sequences of experiences.

Previously, Nuxoll (2007) created agents that used episodic memory to support a variety of capabilities. In that work, agents were given hard-coded procedural knowledge that specified when cues should be created for episodic memory, which structures should be used for cueing retrievals, and how to condition behavior based on the retrieved knowledge. The procedural knowledge was not tuned via learning (such as RL), so the agents *used* episodic memory, but did not *learn to use it*.

In this research, rather than endow agents with preexisting fixed control knowledge, we investigate: learning when to access episodic memory, learning what structures to use as cues, and learning how to condition behavior on the retrieved knowledge. All of these processes are using episodic memory, and this work then learns to use episodic memory in three different senses.

3. Well World

In order to explore how an agent might learn to use an internal episodic memory, we constructed several tasks within an artificial domain we call "Well World." The domain is simple enough to be tractable for an RL agent, but rich enough such that episodic memory can potentially improve performance.

The goal in Well World is to satisfy two internal drives: thirst and safety. Thirst is the agent's primary drive, and it seeks to satisfy that above safety. Thirst is satisfied by consuming water at a well that contains it, while safety is satisfied by consuming the safety resource at the shelter location.

Fig. 1 shows the base Well World environment. In the base configuration, there are three locations. At two of the locations, there are wells; at the third, shelter. In each location, the agent observes a set of attributes and values specific to that location, but does not perceive information

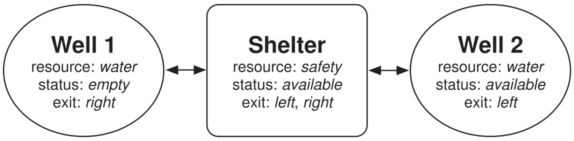

Fig. 1. Objects, resources, and adjacency in Well World.

pertaining to any other location. For example, when the agent is at Well 1, it observes: "location: Well 1", "resource: water", "status: empty". However, it observes no information about the status of the Shelter or Well 2 locations. The agent also observes information regarding exits from a location, but not their destination. For example, at Well 1 the agent observes: "exit: right".

There are two wells which can provide the water resource ("resource: water" in the figure). Well 1 is currently empty, while Well 2 has water available. There is also a shelter, which allows the agent to feel safe when the agent is not thirsty. The shelter always provides the safety resource; it is never exhausted. Water at wells, however, is exhausted after it is consumed once. If water is available at Well 1 and the agent consumes it, the well becomes empty and water then becomes available at Well 2. In this way, the well that provides water alternates every time the agent consumes it.

The agent's drives are modeled as follows. When the agent's thirst is quenched, its thirst drive is 0; thirst linearly increases by 0.1 on every time step. After passing the threshold of 1.0, the agent is considered thirsty until it quenches its thirst, which requires that the agent move to the well object that contains water and then consume water from it, resetting the agent's drive to 0. The agent's drive for safety is constant and cannot be satisfied for longer than the duration of a single action (e.g. the agent is driven to seek shelter on the time step immediately following one in which it has taken an action to satisfy safety). The agent's thirst drive is of primary importance: if the agent is thirsty, it is driven to quench thirst and disregard safety.

Critical to our experimental framework is that the agent is provided with no background knowledge regarding the semantics of the environmental features that it perceives. The agent does not know that if a well is "empty," then it will not be able to drink "water" there; it does not know that "thirst" is a drive that is quenched by "water"; and it does not know about "safety", "shelter", etc. The meanings of all of these features must be learned through trial and error, and the only signal that the agent receives to learn from is reward, the specifics of which are discussed in the next section.

Two of Well World's characteristics make it challenging for RL: first, the agent can only perceive the status of the object in its current location; second, wells alternate in containing water and being empty. To perform optimally, an agent must maintain a memory of the environment (the status of the wells), knowledge that a conventional RL agent lacks.

3.1. Reinforcement in Well World

The reward signal used by an RL agent in Well World is determined by the state of the agent's internal drives, as well as changes in the states of those drives. Reinforcement in Well World is internally calculated by the agent based on its internal drives (similar to Singh, Lewis, & Barto, 2009), rather than determined by the environment as in a conventional RL setting.

The reward values are as follows. There is a cost associated with taking action in the world, which is related to the amount of time it takes to execute an action. As a baseline, actions in an environment incur -1 reward (which is common in RL systems); however, internal actions take (roughly) an order of magnitude less time to complete and thus, they incur -0.1 reward. Since thirst is unpleasant, the agent receives -2 reward on every time step that it is thirsty. As being safe is pleasant, the agent receives $+2$ reward on every time step that it is not thirsty and consumes the safety resource. Satisfying thirst results in $+8$ reward for the agent. Concurrent rewards (e.g. the agent is thirsty and takes an external action) are summed together.

These reward settings have been selected to elicit a certain behavior: namely, the agent should seek water when thirsty and shelter when not. The aspects of the agent's reward structure that are necessary to elicit this behavior include: there is a reward for not staying at the wells when the agent is not thirsty; there is a significant reward for performing the desired action (consuming water when thirsty); and there is a cost for taking external actions and it is greater than the cost of internal actions. Another important property is that there is no explicit reward for using episodic memory, rather the agent must learn control strategies for episodic memory while seeking to satisfy thirst. Changes to the reward structure do not significantly affect the agent's ability to learn to use memory. However, changes to the reward structure can change what the optimal behavior in the task is—as in all RL domains, rewards are a parameter of the environment, not the agent.

4. Experiments in Well World

Within the Well World domain, we developed a suite of four experiments: the first three evaluate various strategies for using episodic memory, while the fourth evaluates strategies for using a simple bit memory mechanism. In the first experiment, we test an agent's ability to learn to select a cue for episodic memory retrieval. The second experiment tests an agent's ability to learn to use the temporal aspects of episodic memory retrievals. The third experiment investigates the agent's ability to create a conjunctive cue (i.e. a cue that contains more than one feature). This set of experiments investigates all of the ways retrievals can access Soar's episodic memory. The fourth experiment tests an agent's ability to learn to use a simple bit memory that does not have all of the functionality of episodic memory, in order to better understand the capabilities afforded by the episodic memory mechanism. Before discussing the experiments and results, we present the details of our agent.

4.1. Agent design and implementation

To explore learning to use episodic memory, we created a Soar agent. In it, procedural knowledge determines what actions can be taken in the external environment as well as what actions can be taken to access the internal episodic memory. On each time step of the environment, the procedural knowledge proposes applicable actions based on the combination of the agent's current perception of the environment and its internal state. It proposes consuming resources that are present, and moving to any objects. There are two internal actions that it can propose for controlling episodic memory (depending on the experiment, as described below): create a cue to initiate a retrieval from episodic memory, or if there has been a retrieval, advance episodic memory forward in time so that the next episode is retrieved. In experiments where the agent must learn which retrieval cue to use, multiple retrieval actions are proposed, one for each cue.

The agent learns how to act in the world via Q-learning (Sutton & Barto, 1998; Watkins, 1989), one of the foundational RL algorithms implemented in Soar (Laird, 2008; Nason & Laird, 2005). As the agent selects internal and external actions, it receives reward, which it uses to adjust its estimates for the value of each action according to temporal-difference updates. The agent's motivation is to maximize future expected reward, and in the experiments below this requires learning to use episodic memory in certain ways.

In the following experiments the agent learns which actions to select in each situation that it is faced with, where a situation is the state of its internal drives, the agent's location, and the features of the well or shelter that it is co-located with. The agent is not learning to generalize or chunk over its procedural knowledge; rather it is learning which actions in the world will result in maximum reward over time. Using RL, the agent learns through trial and error in its environment. The only initial knowledge the agents begin with is the procedural knowledge that proposes possible actions in each state (e.g. consume water, move to shelter, use a particular cue to retrieve an episode from memory).

Action selection is performed using an epsilon-greedy selection process with a linearly decaying exploration rate (Sutton & Barto, 1998). Initially, the agent selects actions with the best value estimates 50% of the time, and the other 50% of the time, it selects from all available actions according to a uniformly random distribution. The random selections prevent the agent from prematurely converging on an action that only initially appears to be best. As time passes, the rate at which random actions are selected decays linearly, until the agent is selecting the actions with the best value estimate 100% of the time. The specific rate of decay differs between experiments, below; it was selected so as to maximize agent learning performance in each task. Selecting random actions is common to agent-based RL algorithms and is necessary due to the exploration/exploitation tradeoff endemic to RL (Sutton & Barto, 1998).

Results presented in this paper are the average of 250 trials. The average values are noisy due to the high underlying variance in reward accumulated at each time step, and therefore the learning curves have been smoothed with a 4253 Hanning function (a standard windowed smoothing function, see Cohen, 1995) and rescaled so that an average reward of 0 per action is optimal in each experiment (the average reward per time step for the optimal policy varies per experiment; this allows for easy comparison between experiments).

4.2. Learning to retrieve episodic memories

The first experiment tests the basic behavior of using RL to learn to use an internal episodic memory. Its purpose is to determine whether an RL agent can learn what to retrieve and when retrieval is appropriate. The agent must learn that when it becomes thirsty it should perform a retrieval from episodic memory, using a cue of "resource: *water*", and then learn which action to take based on the retrieved knowledge.

In Well World (Fig. 1), the optimal behavior in the environment is for the agent to move to the shelter and consume the safety resource when it is not thirsty, and when it is thirsty to move to the well that contains water and then consume it. Agents in Well World are unable to perceive which well contains water, and thus an agent that does not possess an internal memory cannot know which well it must move to when it becomes thirsty. However, an agent endowed with episodic memory can use it to remember which well the agent last consumed water from. The agent's optimal behavior, then, is for the agent to move to the shelter and consume the safety resource when it is not thirsty. When it becomes thirsty, it must select the cue of "resource: *water*" to retrieve the well that it last visited (and hence consumed water from). It then moves to the other well and consumes water there.

Thus, the agent must learn to select specific actions in each situation it encounters. With RL, the agent learns to associate an expected reward with each situation/action pair, and over time, it learns which action for a situation will lead to the most reward. Note that it is not learning the general concept of moving to the opposite well here, as the agent performs no generalization: it must learn both that when it retrieves Well 1 it should move to Well 2 and that when it retrieves Well 2 it should move to Well 1.

Fig. 2 shows the performances of an agent under the following conditions: only the correct cue is available to be learned (labeled "no distracters"); the correct cue and five distracters are available to be learned ("5 distracters"); and a baseline condition in which episodic memory is lesioned and the agent cannot perform retrievals ("lesioned ep. mem."). The baseline condition demonstrates how a "pure" RL agent lacking any internal memory would perform. The five distracter cues are nonsense cues that result in either failed retrievals from memory or episodes in which the agent was at the shelter; retrievals made using those cues are thus not useful for solving the task.

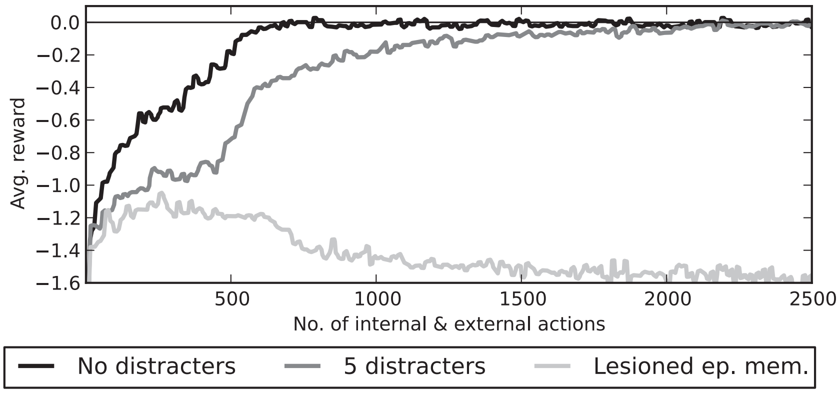

Fig. 2. Performances of agents learning to retrieve episodic memories.

When only a single cue is available for retrieval ("no distracters"), the agent quickly learns both when to act in the environment and when to use its internal memory so as to receive the maximum amount of possible reward. Thus, the agent learns to select actions to satisfy its safety and thirst drives at appropriate times, as described above as the optimal behavior. The presence of the distracters ("5 distracters") slows learning because the agent has to gain sufficient experience in using each cue to learn which cue leads to a retrieval that makes it possible to go to the correct well, and thus higher reward. Finally, when an agent's episodic memory is lesioned, there is never sufficient information in the state for the agent to reliably determine which well contains water when it becomes thirsty, and thus its average reward is low. The results from Fig. 2 indicate that the agent can learn when to use its internal memory while simultaneously learning when to interact with its environment.

4.3. Learning to retrieve what happened next

A unique aspect of episodic memory is that events are linked and ordered temporally. In Soar's episodic memory, memory retrievals can be controlled temporally by advancing to the next memory after performing a cue-based retrieval, providing a primitive envisioning or planning capability where the agent can use its prior history to predict potential future situations. Through RL, the system has the potential of learning when and how to perform such primitive planning.

In the previous experiment, the agent retrieved episodic memories of the last time that it had perceived the water resource, which was sufficient knowledge to determine which well to move to in order to find water. An alternative strategy, explored in this experiment, is for the agent to retrieve a past situation that closely resembles the agent's current situation, and then advance to the next memory to retrieve what the agent did the last time that it was in a similar situation.

In this experiment, the agent has available the normal actions in the environment (moving and consuming resources). It also has two internal actions available to it: a cue-based episodic memory retrieval, which uses all current perceptual knowledge to retrieve the most recent situation that most closely resembled its current situation; and a second action (called *advance*) that retrieves the next episode (the episode that was stored after the episode most recently retrieved). Thus, the agent must learn when to do a cue-based retrieval (where the cue is the complete state), when to advance its retrieval, and what action to take in the world given the knowledge retrieved from episodic memory.

For this task, the optimal behavior for the agent when it is not thirsty is to move to the shelter and consume the safety resource. When it becomes thirsty, the agent must perform a retrieval cued by its current state, which results in the agent remembering the last time it was thirsty at the shelter. The next step is to perform an advance retrieval, which results in the agent remembering where it moved to after it was last thirsty at the shelter. This is followed by moving to the other well, where the agent will find water (as the well that it previously visited will be empty).

An important characteristic of this task is that the knowledge stored in episodic memory and the agent's actions in the world are more closely related than in the previous experiment. The best strategy for memory usage strongly depends on the agent's prior actions in the environment; if the agent does not visit and consume resources in the appropriate order (i.e. behave optimally for external actions), then the agent is not guaranteed to gain useful information from internal memory retrievals. In the previous task, a successful retrieval contained knowledge of the last well from which water was consumed. In this task, a successful retrieval contains knowledge of the action that was taken after the agent last became thirsty: this action may not be informative if the agent has not yet learned a good behavior.

The performances of the agent under two conditions are plotted in Fig. 3. In the first condition, the agent learns when to make a cue-based retrieval and when to advance the episodic memory, although the advance action cannot be selected until a cue-based retrieval has taken place (this

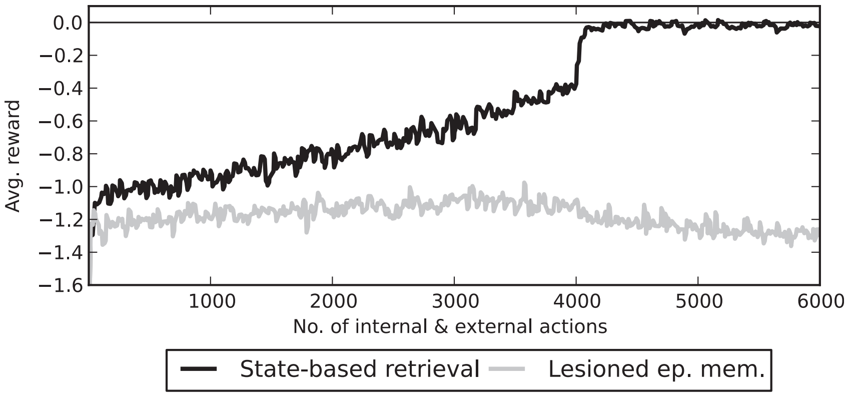

Fig. 3. Performances of agent using temporal control of episodic memory after retrieval.

condition is labeled “state-based retrieval”). The second condition is a baseline comparison in which episodic memory is lesioned.

There is a dramatic improvement in performance after the agents have taken 4000 actions. In this experiment, the exploration parameter decays linearly over time (as discussed in Section 4.1). On the 4000th action, and every successive action after that, the agent ceases to select actions randomly and only selects the actions with the best learned value estimates. The dramatic difference in performances between when the agent is selecting random actions very rarely (e.g. just before the 4000th action) and never (e.g. immediately after the 4000th action) indicates that small amounts of random action selection in this task have a significant negative impact on behavior. In the previous task, random action selection did not interfere with an agent’s ability to recover and complete the task.

Our hypothesis for this sensitivity is that in this task, there is a precise sequence of actions that must be executed. If an agent becomes thirsty and randomly selects to consume the safety resource, then when the agent next becomes thirsty and attempts to retrieve a memory of when it was last thirsty and what it did next, that memory will not be informative as to which well contains water – which in turn leads the agent to bias its learned behavior against performing retrievals. Effectively, any random action selection is disruptive when episodic memory is used to remember a sequence of past actions.

Another notable feature of the results is that while the agent nearly reaches the optimal level of performance, the agent does not converge to the optimal behavior in all trials, but instead converge to a behavior that is very near to optimal. We determined that the agent achieves optimal behavior 71% of the time.

In the cases where the agent converges to the sub-optimal behavior, it is not using episodic memory retrievals to recall a past situation and then advance to the next situation. Instead, it is using episodic memory in a more primitive way, as a single bit of information, as was used in the agents in Littman (1994) and Pearson et al. (2007), as well as our fourth experiment presented in Section 4.5. In this second behavior, when the agent becomes thirsty, it immediately moves to one of the wells (the same well every time). If the well contains water, it consumes it; if not, it performs a cue-based retrieval and moves back to the shelter. At the shelter, the agent now knows that it has performed a retrieval and instead of moving to the same well again (the one that it just visited and knows is empty), it moves to the other well and consumes water there, regardless of the contents of the retrieval. Essentially, the agent learns which well to move to when it is thirsty based on whether a retrieval has been performed, and not based on the contents of what was retrieved. This behavior is suboptimal because it requires an additional action in the environment every time the agent becomes thirsty, leading to slightly less accumulated reward over time.

These phenomena are explained by the difficulty of the learning problem that was identified above: for the agent to learn the optimal strategy for using its internal memory, it must also learn a near optimal strategy for acting in the environment. The learning problem is difficult because the effects of the agent’s memory actions depend on the history of the agent’s actions in the environment, which the agent cannot perceive (technically, the problem is *partially observable*; Sutton & Barto, 1998). The agent must learn how to use its memory while settling on a good behavior in the environment, but it must also settle on a good behavior in the environment without knowing how to use its memory. Often the agent is successful in learning to simultaneously control both memory and external action, but occasionally the agent is unable to converge to the best behavior.

4.4. Learning to construct a retrieval cue

In the first experiment, one condition involved the agent learning to select between multiple cues when retrieving from memory. In the second experiment, the agent used cues with more than one feature (features of its current state) in order to retrieve from memory. The purpose of the third experiment is to investigate whether an agent can learn to select multiple features to use as a cue, combining aspects of both previous experiments.

In order to test this capability, it was necessary to extend the base Well World configuration so that there were more wells and more features that could be used for retrieval. A third well was added to the environment, and a color feature was added to all objects; the modified environment is shown in Fig. 4. As in the base environment, only Wells 1 and 2 ever contain water, and they continue to alternate between full and empty as before. Well 3 never contains water; it was added to the environment to serve as a distracter to the agent when it performs a cue-based retrieval with features not present on the other two wells.

In this task, the optimal behavior when the agent is not thirsty is still to navigate to the shelter and consume the safety resource. When thirsty, the agent must construct a cue containing features corresponding to the two wells that can contain water in order to determine which well it visited last; these features are “resource: water” and “color: blue”. After retrieving the memory of the last blue well that it visited, the agent must then navigate to the *other* blue well and consume water there to satisfy its thirst. To achieve the effect of a cue with multiple features, the agent performs successive cue-based retrievals from episodic memory (e.g. performing a cue-based retrieval with “resource: water”, then a second retrieval with “color: blue”, creates a cue of the conjunction of those features).

If the agent constructs a cue with some other combination of features, the retrieved episode does not provide sufficient information for the agent to determine which well to visit next, since no combination of features result in a memory of the well that last contained water. Soar’s episodic memory mechanism retrieves the most recent episode when multiple memories are perfect matches to the cue, thus building a cue that contains only “resource: water” or “color: blue” does not result in the agent remembering the last blue well that it visited (assuming that it has moved back to the shelter). Instead, “color: blue” leads to the retrieval of the shelter, while retrieval of “resource: water” leads to retrieval of Well 3.

The performances of the agent that constructs retrieval cues in the modified Well World are shown in Fig. 5 for three conditions: learning to construct a cue from only the two correct features (“no distracters”), learning to construct a cue when two distracters are also present, and a baseline where episodic memory is lesioned. In the first two conditions, there are different sets of features with which an agent may construct the cue: the first has only the two correct features available (resource: water, and color: blue), while the other also has their complements (resource: water/shelter, and color: blue/red). A cue can be any subset of the available features, and thus the agent must learn to construct the correct cue in both cases.

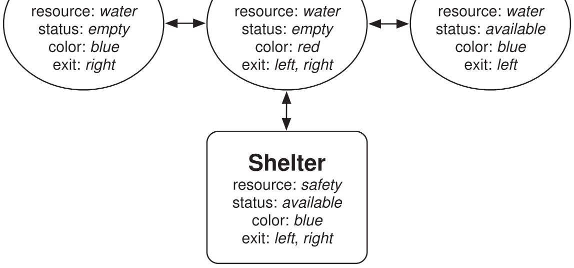

Fig. 4. Well World modified with an additional well and an additional feature, color.

The agent converges to the optimal behavior under both conditions, more slowly when two distracter features are present, as expected. These results indicate that an agent can learn to build conjunctive cues from primitive features, and use them in a task to retrieve from episodic memory.

4.5. Learning to use bit memory

The three experiments above demonstrate that there are circumstances in which it is computationally feasible to learn to use episodic memory. The Well World domain, however, is relatively simple and in order to perform successfully in it the agent must remember only a single unit of knowledge: the identity of the well from which it last consumed water. Given that only a single unit of knowledge is necessary in order for an agent to perform optimally, it might be computationally feasible for an agent endowed with a less powerful memory mechanism to learn behaviors comparable to those learned with episodic memory. The experiment in Section 4.3, in which the agent learned to use episodic memory as a bit memory on some trials, directly motivated this experiment.

Episodic memory is more powerful than necessary in the Well World domain because of several characteristics. First, episodic memory has unlimited capacity, whereas performance in Well World requires only a single unit of knowledge. Second, knowledge in episodic memory has unlimited persistence, whereas the Well World domain requires that the identity of the last well that was consumed is known only for a finite number of steps in the domain, after which the knowledge becomes irrelevant to future behavior. Third, knowledge is stored to episodic memory automatically via an architectural mechanism, whereas an agent could learn to store knowledge deliberately instead. Fourth, knowledge in episodic memory is accessed via a retrieval cue, allowing for single episodes to be retrieved and thus condition immediate behavior.

In order to better understand the dynamics of episodic memory and reinforcement learning in Well World, we created an agent endowed with a simple bit memory mechanism. Comparing the behaviors of agents that learn to use the different mechanisms allows us to better understand whether some or all of these characteristics of episodic memory are essential for an agent to be capable of learning to use memory while acting in Well World, or whether simply being endowed with a simpler memory mechanism is sufficient.

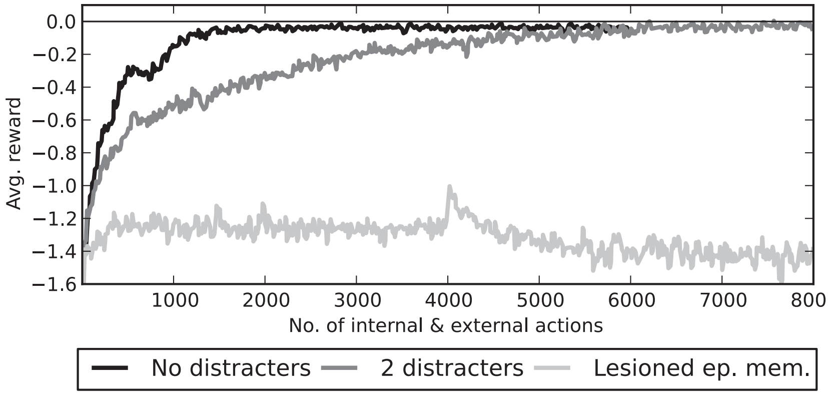

Fig. 5. Performances of agents that construct cues with more than one feature for episodic memory retrievals.

The bit memory mechanism that we used for this experiment consists entirely of a single unit of knowledge in Soar's working memory that is under the agent's deliberate control. The agent can set the contents of memory to either *true* or *false* by selecting corresponding internal actions, and it can then use the knowledge stored in bit memory to condition its behavior in Well World. This bit memory mechanism is inspired by earlier work that investigated learning to use memory with simple internal memory mechanisms (Littman, 1994; Peshkin, Meuleau, & Kaelbling, 1999).

The performance of the agent endowed with bit memory is evaluated in the original Well World domain (Fig. 1) under three different conditions. In the first condition, the agent does not need learn a memory management strategy, but rather has a fixed strategy in which it toggles its bit memory to *true* after consuming Well 1 and to *false* after consuming Well 2. This agent learns to use the contents of bit memory in order to support correctly select actions in the environment, such as moving and consuming resources. In the second condition (called "Fixed true strategy"), the agent has a fixed strategy for when to set its internal memory bit to *true* (after consuming Well 1), but must learn when to set its memory bit to *false* as well as how to act in the environment. In the third and final condition, the agent has no fixed strategy to control its memory, and instead must learn both when to toggle its memory as well as how to act in the environment.

The performances of the agent under these three conditions is illustrated in Fig. 6. The agent that had a fixed strategy for managing its bit memory learns very quickly how to perform in Well World, converging to the optimal behavior in 500 actions. The performances of the other two agents are both poor and nearly equivalent, although the agent that has a partial memory management strategy behaves slightly better than the agent that has no fixed memory management strategy between the 250th and 1000th actions.

That the agent with a fixed memory management strategy quickly converges to the best possible behavior in the domain demonstrates that a single bit of memory is sufficient in order to support optimal behavior in the Well World setting. However, both agents that had to learn strategies to control their bit memory failed to learn effective behaviors, indicating that in Well World agents cannot learn to use bit memory.

5. Discussion

We begin by discussing the three experiments in which an agent learned to use episodic memory while simultaneously learning to perform in the environment. Although

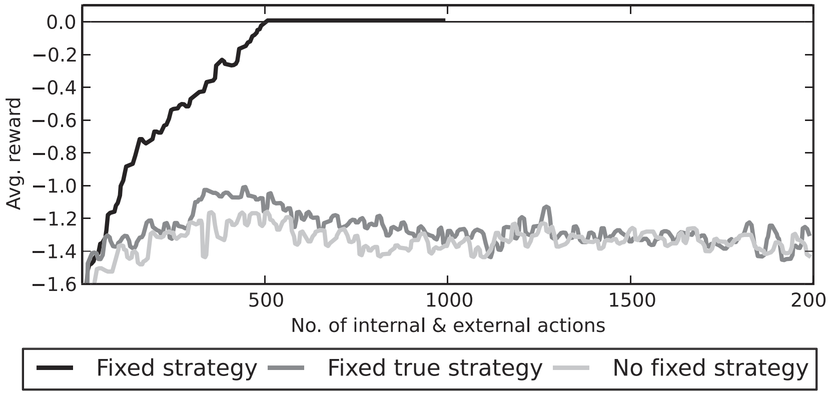

Fig. 6. Performances of agents learning to use bit memory in Well World.

the agent is faced with learning to use its memory while acting in the environment (and thus affecting what knowledge will be retrieved from memory in the future), the interaction of memory and action in the environment is significantly more intertwined in the second experiment. There, the agent's past actions directly impact the utility of knowledge retrieved from episodic memory. In all experiments, the agent learns very early on to consume the safety resource when it is not thirsty, and to immediately move to the shelter as soon as it is not thirsty. In the first and third experiments, this means that when the agent retrieves an episode from memory using features of a well as a cue, it will typically be the well that it last consumed water from. However, in the second experiment, the agent is retrieving memories of the first action that it took to quench its thirst, and *not* the memory of when it finally managed to quench it. It not only takes longer to learn how to best act in this setting, but the eventual result is that it sometimes converges to a local maximum in the behavior space instead of converging to the globally optimal behavior.

These three experiments demonstrate that RL can be applied successfully to learn to use internal actions over an episodic memory mechanism while simultaneously learning to act in its environment. Additionally, RL alone cannot be successfully applied to those same tasks, demonstrating that there is a functional advantage to combining RL with an episodic memory in some settings. We also demonstrated that RL can learn when to retrieve, learn which cue to use for retrieval, learn when to use temporal control, and learn to build a cue from a set of possible features.

In the fourth experiment, it was demonstrated that while a bit memory can store sufficient knowledge for an agent to act in Well World, an agent endowed with bit memory is unable to learn to use it while learning to act in the domain. This contrasts with results from the first three experiments, in which it was demonstrated that an agent can learn to use episodic memory. It starkly contrasts with the results from Section 4.3, in which an agent did learn to use a more expressive episodic memory as a less powerful bit memory.

Although episodic memory is powerful and significantly more complex than bit memory, the process by which retrievals are made in these experiments tightly constrains the space of possible actions that are available to an agent. In the first and third experiments, a small and finite number of possible cues are available with which an agent can make retrievals from episodic memory. This has the effect of limiting the amount of initial learning that an agent must undertake before finding effective memory usage strategies.

When using bit memory, however, the space of memory control strategies is relatively under constrained. When the agent has no fixed strategy for memory management, the space of possible memory usage strategies is so large that it is unable to find an effective strategy. Even when the agent has a fixed strategy for when to toggle its memory bit to *true*, the agent cannot effectively settle on a strategy for when to toggle its bit to *false*, simply because if the agent toggles it at the wrong time it cannot simply toggle it back since it has no other knowledge that might help it recover from an error.

6. Conclusion

More broadly, this research opens up the possibility of extending the range of tasks and behaviors modeled by cognitive architectures. To date, scant attention has been paid to many of the more complex properties and richness of episodic memory, such as its temporal structure or the fact that it does not capture just isolated structures and buffers but instead captures working memory as a whole. Similarly, although RL has made significant contributions to cognitive modeling, it has been predominantly used for learning to control only external actions. This research demonstrates that cognitive architectures can use RL to learn more complex behavior that is dependent not just on the current state of the environment, but also on the agent's prior experience, learning behavior that is possible only when both RL and episodic memory are combined.

Although our research demonstrates that it is possible to learn to use episodic memory, it also raises some important issues. Learning is relatively fast when the possible cues lead to the retrieval of an episode that contains all of the knowledge that an agent requires in order to determine how to act in the world. When retrieving episodes that most closely match the current state and then using temporal control of memory to remember what happened next, however, learning is slower and does not always converge to the best possible behavior. Learning to use episodic memory to project forward is difficult – requiring many trials to converge and without a guarantee that optimal behavior will be achieved. Do these same issues arise in humans or do they have other mechanisms that avoid these issues? One obvious approach to avoid the issues encountered in our experiment is to use one method, such as instruction or imitation, to initially direct behavior so that correct behavior is experienced and captured by episodic memory, and then learning to use those experiences would probably be much faster.

Another approach that we are pursuing is to simplify both the memory models and tasks to better understand how characteristics of each influence an agent's potential to learn to use memory. Although the tasks presented in this paper appear to be simple, they do contain a number of features and rewarding situations. Soar's episodic memory mechanism is complex and powerful. By investigating agents endowed with simpler memory models situated in the simplest of tasks, we can incrementally add different sources of complexity and measure the effects. This approach is directly motivated by our desire to better understand the implications of our fourth experiment, in which we demonstrate that an agent cannot learn to use bit memory while it can learn to use episodic memory.

Acknowledgments

The authors acknowledge the funding support of the Office of Naval Research under Grant Number N00014-08-1-0099. We thank Andrew Howes, Richard M. Young, and anonymous reviewers of Cognitive Systems Research and the 9th International Conference on Cognitive Modeling for their insightful and thoughtful comments.

References

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York, NY: Oxford U. Press.

Cohen, P. R. (1995). *Empirical methods for artificial intelligence*. Cambridge, MA: MIT Press.

Derbinsky, N. & Laird, J. E. (2009). Efficiently implementing episodic memory. In *Proceedings of the 8th international conference on case-based reasoning.* Seattle, WA.

Laird, J. E. (2008). Extending the soar cognitive architecture. In *Proceedings of the first artificial general intelligence conference* (pp. 224–235). Memphis, TN.

Littman, M. L. (1994). Memoryless policies: Theoretical limitations and practical results. In *Proceedings of the 3rd international conference on simulation of adaptive behavior*. (pp. 238–245).

Nason, S., & Laird, J. E. (2005). Soar-RL, integrating reinforcement learning with soar. *Cognitive Systems Research, 6*, 51–59.

Nuxoll, A. (2007). *Enhancing intelligent agents with episodic memory.* Doctoral dissertation, Computer Science & Engineering, U. of Michigan, Ann Arbor.

Nuxoll, A. & Laird, J. E. (2007). Extending cognitive architecture with episodic memory. In *Proceedings of the 21st national conference on artificial intelligence.* Vancouver, BC.

Pearson, D., Gorski, N. A., Lewis, R. L. & Laird, J. E. (2007). Storm: A framework for biologically-inspired cognitive architecture research. *ICCM-07*. Ann Arbor, MI.

Peshkin, L., Meuleau, N., & Kaelbling, L. P. (1999). Learning policies with external memory. In *Proceedings of the 16th international conference on machine learning*. Bled, Slovenia.

Singh, S., Lewis, R. L., & Barto, A. G. (2009). Where do rewards come from? In *Proceedings of the annual conference of the cognitive science society.* Amsterdam.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.

Tulving, E. (1983). *Elements of episodic memory*. Oxford: Clarendon Press.

Wang, Y. & Laird, J. E. (2006). *Integrating semantic memory into a cognitive architecture*. (Tech. Rep. CCA-TR-2006-02). Ann Arbor, MI: Center for Cognitive Architectures, University of Michigan.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. Doctoral dissertation, Cambridge University, Cambridge, England.

Zilli, E. A., & Hasselmo, M. E. (2008). Modeling the role of working memory and episodic memory in behavioral tasks. *Hippocampus, 18*, 193–209.