Extracting Episodic Memory Feature Relevance Without Domain Knowledge

Benjamin WALKER, Dustin DALEN, Zachary FALTERSACK, Andrew NUXOLL
University of Portland 5000 N. Willamette Blvd. Portland, OR 97203

Abstract. Episodic memory provides many important capabilities to a cognitive architecture. One of the challenges of creating a general episodic memory system is to be effective when given no information about the agent’s task.

In this paper, we present an effective algorithm for detecting the relevance of the features of episodic memories while only being told when an agent completes a goal. We demonstrate this algorithm using an episodic learning agent in a task that provides the agent with a mix of relevant and irrelevant features. The episodic learner outperforms a variant Q-Learning algorithm that has proven effective in the past.

Keywords. episodic memory, feature selection

Introduction

Human episodic memory is a long-term memory of specific events from an individual’s experience [15]. Episodic memory is typically distinguished from semantic memory and procedural memory in that the episodic memories, or episodes, include a specific temporal component that links them to a single, specific event. In contrast, semantic memories contain facts, things you know but can not recall when you learned them. For example, knowing that the United States capital is Washington DC or that cherries grow on trees. Procedural memories are things you know how to do but can not articulate such as whistling or juggling.

Impairment of one’s ability to encode or retrieve episodic memories is the condition known as amnesia. Studies of severe amnesiacs [12] provide evidence that episodic memory is a distinct, long-term memory system as amnesiacs retain their ability to create and retrieve semantic and procedural memories. Studies show that the hippocampus is involved in the encoding and retrieval of episodes and that the frontal cortex is likely where such memories are stored in the long term [16].

One effect is clear: amnesia is a severe handicap of an individual’s cognitive ability. This implies that episodic memory is an essential component of an effective cognitive architecture.
Feature Relevance

A critical component of an effective episodic memory system is the ability to retrieve the most helpful memory for a given task. If an agent is using its episodic memory to inform its decisions, then poor retrievals clearly result in degraded behavior.

However, determining the “best” match for a given memory cue is difficult in the general case. Consider two agents that process information in an online bookseller’s database. One agent makes suggestions to customers based upon their past purchases. The other agent selects appropriate shipping boxes for books sold by an online bookseller. For either agent, the bookseller’s database might contain a lot of information that is irrelevant to the agent’s task.

Typically, an agent created specifically for a purpose would be given information about what features of the database are most and least relevant to the task. A general cognitive architecture does not have this luxury; by definition, it must be effective in any environment. Therefore the episodic memory system that is integrated into a cognitive architecture must rely upon the agent’s behavior to induce the relative importance of each feature of a given memory cue.

All of the existing episodic memory systems we are aware of rely upon being given knowledge about the relative importance of various features from an external source. In the context of a general episodic memory system, a feature is any component of the memory that is used for matching during retrieval. Most systems require the agent to identify important features of episodes by only using such features when it constructs a memory cue [1,7,14]. One system extracts that information indirectly from the agent’s domain-specific behavior [11].

In either case, this means that if the agent does not have knowledge about the task it is performing the episodic memory system also lacks this knowledge. Presuming our goal is create a cognitive architecture to support general intelligence, it is desirable to create an episodic memory system that can gauge the importance of each feature of an episode intrinsically.

The problem of selecting a small subset of relevant features from a larger whole is known as feature selection or feature weighting. (See [5] for a survey). This topic has received decades of attention among artificial intelligence and data mining researchers. However, much of the current body of feature selection research is not directly applicable to cognitive architecture design because it assumes that feature weighting is inherently a supervised-learning task.

Feature selection for model-free reinforcement learning has not been ignored but the research we have found to date [6,8,9] assumes that the domain that the agent is operating in has the Markov property. Specifically, it assumes that the agent’s current state contains all the information necessary to make an optimal decision. Clearly a cognitive architecture can’t make this assumption. Furthermore, tasks that require the agent to have an episodic memory of past events clearly lack the Markov property.

Thus, the problem addressed by this paper is how to weigh the importance of the various features of an episodic memory when the agent is given minimal information about the environment. We describe this task using the term feature relevance rather than feature selection or feature weighting. This is because the
former term implies the omission of features from the data and the latter term has a broader definition associated with general nearest neighbor methods.

**Episode and Action Based Feature Relevance**

In this section, we outline a simple algorithm we’ve devised for determining feature relevance in this general case.

Our algorithm makes the following assumptions about the episodic memory system. These assumptions are compatible with the domain-independent, artificial episodic memory systems we are aware of:

- The episodic memory system records a new episode each time the agent takes an action in its environment.
- The episode contains a snapshot of some portion of the agent’s current sensing, current goal and internal features determined via cognition.
- The agent selects each action from a finite set of actions \( A_1, A_2, \ldots, A_n \).
- The agent attempts to accomplish one goal at a time and is aware of what goal it is attempting to achieve. Each goal has exactly one goal state associated with it.
- The agent is aware when it has successfully completed a goal.
- Upon achieving a goal, the agent will immediately begin working toward achieving a subsequent goal.

In brief, our algorithm identifies clusters of episodes that should represent the same state because they began after the same goal and were arrived at via the same actions. The algorithm then weights the features of these episodes to minimize the distance between each episode in a binary \( N \)-space defined by the presence or absence of their features in a manner much like a weighted nearest neighbor match [3] except in reverse. In other words, the algorithm’s goal is to weight the \( k \) features of related episodes so that they are as close together as possible in the \( k \)-dimensional space.

Our algorithm in more detail is as follows:

**Step 1:** Extract the set of all past runs to the goal that have begun after achieving the agent’s current goal.

**Step 2:** Extract the subset of runs that begin with the most commonly taken action. For example, if the agent took action \( A_3 \) more often than any other, then extract all runs whose first action was \( A_3 \).

**Step 3:** Examine the features of all the episodes in the subset that resulted from step 2. For each feature, identify what fraction of the time that feature appears in all the episodes in the subset (a value in the range \([0.0..1.0]\)).

**Step 4:** Recursively repeat steps 2 and 3 on the subset of runs for subsequent actions in the sequence. Each time, extract a new, smaller subset of runs whose next action is the most commonly taken in the subset selected the previous time. Stop when the subset’s size is less than 2.

**Step 5:** Weight each feature based upon how much more or less frequently it appears in the extracted subset compared to the overall average for all the episodes in the episodic memory. Thus, if a feature typically appears 30% of the
time among all episodes, but appears 70% of the time in the subset then the feature receives a high weight. If, instead, the feature appeared 40% of the time in the subset it would receive a much smaller weight. Alternatively, if the feature appeared 5% of the time in the subset then its absence receives a high weight.

The weights that result from this algorithm can be used as required by the episodic memory system. The most obvious purpose would be to weight the features of a memory cue that is used for retrieval.

**Blind Navigation Domain**

To test the effectiveness of our algorithm we created a domain wherein the agent attempts to reach a goal position within a simple two-dimensional maze. This domain is a virtual implementation of a robotic navigation environment we have created using iRobot Create robots [2]. This environment is, in turn, a more challenging version of one that was used to demonstrate the NSM Q-Learning algorithm developed by McCallum [10].

In this domain, the agent is provided with a very limited set of binary sensors:

- **left bumper**: detects when the agent bumps into a wall on its left
- **right bumper**: detects when the agent bumps into a wall on its right
- **goal**: detects when the agent has reached the goal position

It’s notable that if the agent hits a wall dead on, that both the left and right bumper sensors will activate.

The agent has a few simple actions it can take at any given time step:

- **forward**: move forward a short, fixed distance
- **left**: turn left 45 degrees
- **right**: turn right 45 degrees
- **no-op**: three additional “noise” actions that each have no effect

Because of the agent’s highly limited sensors, state aliasing is common. As a result, an episodic knowledge of past states is essential for making good decisions in the present. Hence, the agent requires an episodic memory to complete this task.

We want to emphasize that the agent is not informed of the nature of its task. The meaning of its sensors and actions are not communicated in any way other than identifying the goal sensor. Thus, the agent is only aware that it must accomplish a goal with the given inputs and outputs.

This enforced ignorance makes the domain intensely difficult rather than trivial. In recent experiments with human subjects (not yet published) only about 70% of them were able to complete the simplest maze we could define in this domain.

To test the effectiveness of our feature relevance algorithm we modified the environment to introduce seven additional noise sensors. At each time step, the environment chose randomly whether to include or omit each of the associated features. These features appear to the agent as additional binary sensors, indistinguishable from the ones defined above.
An Episodic Learning Agent

To test the system, we created an episodic learning agent that uses an episodic memory system we’ve developed called Ziggurat [4]. This system is able to overcome the state aliasing issues associated with our blind navigation domain by recognizing sequences of past actions that tend to be associated with each other and also using those sequences as part of its match process.

At each time step, the agent uses its episodic memory to select its actions. It does this by searching its memory for the most similar past situation in which it took the same action. Then, it builds a plan to reach its goal as if it has returned to that past state again. Next, it follows the plan until it fails or the goal is reached. Finally, it repeats the above procedure until the simulation ends.

We modified this baseline agent to bias its episodic retrieval match using the weights derived using the feature relevance algorithm we described above. Specifically, it uses its entire sensing as a cue, but then uses the feature weights to prune its cue to only those features above a certain, fixed threshold. For our experiments, we selected a threshold of 15% more or less likely. Other reasonable threshold values seemed to perform about equally well in preliminary experiments.

Experimental Results

To evaluate the performance of our agent we compared it to an agent using the NSM Q-Learning algorithm. This algorithm was originally demonstrated using a similar domain [10]. This algorithm also uses a rudimentary episodic memory to overcome state aliasing and performed well in the domain before the noise features were added.

For our experiment, we used a U-shaped maze with the start and goal positions at opposite ends of the ‘U’. We ran each agent from start to goal 100 times in this maze. Each time, the agent retained episodic memories of previous runs to the goal and, thus, could leverage those memories to improve performance.

The experiment was repeated 350 times and the results were averaged. Figure 1 depicts the average number of steps to the goal for each agent on each of the 50 runs. The horizontal line depicts the average performance of an agent that always selects a random action.

By examining the figure it is apparent that, with the additional noise features active, the NSM Q-Learning agent is no longer able to perform better than a random agent in this maze.

However, the episodic learning agent is able to use the feature weights to gradually learn which of its sensors are relevant to that task. This knowledge, in turn, allows the agent to learn to navigate the maze more quickly.

To confirm our results, we made a direct examination of the feature weights at the conclusion of these runs and found that they accurately reflected the relevance of the sensors. We also verified that the episodic learning agent was incapacitated when the feature weighting was removed.
Figure 1. This graph depicts the average number of steps required to reach the goal on 100 successive trials with each of the three agents.

Discussion

The task we have defined in this research is that of identifying the task relevance of the features of an episodic memory given no advance information about the agent’s task and making little or no presumptions about the task. We have also argued that feature relevance is an important part of a general episodic memory for a cognitive architecture.

In this paper, we have presented an algorithm for calculating feature relevance in this situation that is based on nearest neighbor match algorithms. We’ve used the relevance values to weight episodic memory retrieval in an episodic learning agent.

While simple, our algorithm is the only one we know of that can learn feature relevance in this general case. However, the general feature relevance problem is particularly challenging and we can identify two key weaknesses that need to be addressed for greater success. First, we have only demonstrated this algorithm with a single domain. Changes in the domain and to the recorded episodes over time would have an unknown effect on performance. Second, our algorithm relies upon having information about the agent’s current goal and being certain when that goal is achieved. For a general reinforcement learning task, the agent may only be given reward information rather than goal information.

For the future, we want to explore alternative algorithms that do not rely upon being aware of the agent’s current goal. We also intend to apply feature relevance algorithms in a broader range of domains in order to provide a more compelling argument for the algorithm’s generality.
References


