Improving the Learning Rate by Inducing a Transition Model

Robert Bridle  
Department of Computer Science  
The Australian National University  
rbridle@cs.anu.edu.au

Eric McCreath  
Department of Computer Science  
The Australian National University  
ericm@cs.anu.edu.au

Abstract

In general, a reinforcement learning agent requires many trials in order to find a successful policy in a domain. In this paper we investigate inducing a transition model to reduce the number of trials required by an agent. We discuss an approach that incorporates transition model learning within a contemporary agent design.

1. Introduction

Reinforcement learning can be an appropriate means of enabling agent adaptivity, since both are characterised as learning through the exploration of interactions with an environment [7]. However, reinforcement learning may not be appropriate if an agent cannot freely interact with its environment. In this paper we address this limitation by learning a model of the agent’s environment.

Typically, an agent could apply a Q-learning technique to solve a reinforcement learning task [8]. However, Q-learning requires many interactions with an environment. Also the Q-function implicitly combines both the transition model and the distance from the goal. This makes it a more complex function to represent and learn than if the components are considered independently.

Previous attempts to learn a transition model of an agent’s environment, such as Sutton’s Dyna [6] and Lin’s QCON and AHCON [2] frameworks have been unable to capture in training examples and possible hypotheses the complex structure of an environment. We propose using a language capable of capturing this complex structure to learn a transition model, which specifies the state an agent will be in after performing a given action in a given state. The transition model will reduce the reinforcement learning task to a process approximating dynamic programming [1].

We examine and evaluate two methods of approximating the transition model with two Q-learning methods. Also a description of an address book agent that incorporates the concept of transition model learning is presented.

2. Learning the Transition Model

The first method of approximating the transition and reward function uses ALKEMY, a high-order decision tree learner [3]. ALKEMY classifies state action pairs according to how the action affects the state. The second method uses FJLIME, which forms functions from individual examples of an agent’s transition. These functions are combined into a single function that works for all examples.

A Tileworld [4] environment was used to compare the ALKEMY and FJLIME transition model learning agents, with two Q-learning agents. The first Q-learning agent uses a tabular representation of a Q-function, and the second approximates the Q-function using ALKEMY as a regression tree learner.

The Q-learning agents follow a policy specified by their Q-function which results in either a successful or unsuccessful trial in the environment. The transition model learning agents attempt to discover successful policies through a breadth-first search of their transition model. Each transition in a trial is recorded and then used to update the agent’s transition model.

After 100 trials in the Tileworld environment, the learning-rate of the transition model learning agents was an order of magnitude higher than the Q-learning agents. This difference can be explained by the complexity of learning the Q-function when compared to the transition model. Also, the transition model learning approach allows all successful policies to be found, while the Q-function only encodes the optimal policy.

3. Using the Transition Model

We now present our findings on incorporating transition model learning into an address book agent. The agent’s percepts are the statements and queries made by the user, and the agent’s actions are the responses given to the user. The
user also provides feedback to the agent, either by accepting the agent’s responses or by responding with “Thanks” or “No”. The agent must learn appropriate actions to achieve positive feedback from the user.

This is now illustrated with an example. Suppose the user asks for John’s phone number. If the agent is unable to uniquely identify the name of the person, it will have to ask the user who they are talking about. The next time the user asks for John’s phone number, in a similar context, the agent could use the previous interaction to learn which John the user wants. This interaction is seen in Figure 1.

![Image](Figure 1. Example of the address book learning that John refers to John McCreat when requesting the phone number.)

The current implementation of the address book agent is based upon the AgentSpeak language [5] and is intended to incorporate transition model learning within the BDI model of agency. In AgentSpeak the behaviour of the address book agent is governed by a set of plans. A plan is triggered by a statement or query from the user. A triggered plan is executed if its context can be satisfied. The intent of a plan is to correctly respond to the user’s query or statement in a given situation. A method for incorporating transition model learning into the AgentSpeak language is through the addition of new plans that are based on modifications of existing plans. If the address book agent can determine the reason why a user responds in a similar way to an existing plan, then this knowledge is used to create a new plan by modifying the existing plan’s context. The new plan would be applicable only in this modified context. The new plan’s action would reflect the action that would have come about had the user specified a response. The creation of the new plan through the modification of an existing plan is shown in Figure 2.

A partial transition model in the form of additional plans applicable only in the correct context are used to reduce interactions with the user.

## 4. Conclusion

We have presented an approach for inducing a transition model in reinforcement learning. A description of an address book agent that learns users’ preferences with minimum interaction illustrates a fitting application for transition model induction in reinforcement learning.

### Acknowledgements

The authors would like to acknowledge the financial support of the Smart Internet Technology CRC.

### References