Learning State Features from Policies to Bias Exploration in Reinforcement Learning

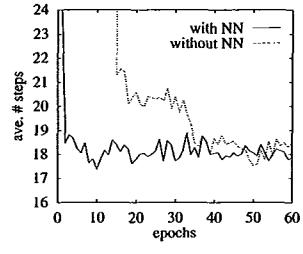
Bryan Singer and **Manuela Veloso**
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213
Email: {bsinger+, mmv+}@cs.cmu.edu

When given several problems to solve in some domain, a standard reinforcement learner learns an optimal policy from scratch for each problem. This seems rather unfortunate in that one might expect some domain-specific information to be present in the solution to one problem for solving the next problem. Using this information would improve the reinforcement learner's performance. However, policies learned by standard reinforcement learning techniques are often very dependent on the exact states, rewards, and state transitions in the particular problem. Therefore, it is infeasible to directly apply a learned policy to new problems, and so several approaches have been and are being investigated to find structure, abstraction, generalization, and/or policy reuse in reinforcement learning (e.g., as overviewed in (Sutton & Barto 1998)). Within our line of research, we describe each state in terms of local features, assuming that these state features together with the learned policies can be used to abstract out the domain characteristics from the specific layout of states and rewards of a particular problem. When given a new problem to solve, this abstraction is used as an exploration bias to improve the rate of convergence of a reinforcement learner.

There are two assumptions required by our learning approach: (i) a domain with local state features predefined for each state, and (ii) a set of sufficiently simple Markov Decision Problems (MDPs) within the domain. Our approach consists of the following procedure:

1. Using a Q-learner, we solve a set of training problems, saving the Q-tables.
2. We generate training examples from the Q-tables by describing states by their local state features and labeling the examples as either positive or negative for taking a particular action. Specifically, the action that has the maximum Q-value out of a state is used as a positive example, as it is viewed as one that may be worth exploring in other similar states. Likewise, an action that has a Q-value indicating that this action never led to the goal state is one that may not be worth exploring in other similar states and is labeled as a negative example.
3. We train a set of classifiers on the examples to map local state features to successful actions. Each classifier in the set learns whether a particular action will or will not be successful.
4. In new problems, we bias the reinforcement learner's exploration by the trained classifiers. This bias allows the reinforcement learner to solve other, potentially more difficult, problems within the domain more effectively than a reinforcement learner would if starting from scratch.

We have empirically validated this algorithm in grid-like domains, and in particular, in the complex domain of Sokoban (Wilfong 1988). Sokoban is an interesting domain of puzzles requiring an agent to push a set of balls to a set of destination locations without getting the balls stuck in various locations such as corners. From a set of training puzzles, we trained a neural network to map from features describing the locations a small radius around the agent to successful actions. As an example of our results, the figure shows the average number of steps required to reach the goal in a new puzzle during learning for both a Q-learner biased with the learned neural network and an unbiased Q-learner. We found that by utilizing solutions to previous problems, our algorithm significantly reduces learning time in new problems.



Acknowledgements Special thanks to Sebastian Thrun for several initial discussions. This work is sponsored in part by DARPA and AFRL under agreement number F30602-97-2-0250 and by a NSF Graduate Fellowship.

References

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction.* MIT Press.

Wilfong, G. 1988. Motion planning in the presence of moving obstacles. In *Proceedings of the Fourth Annual ACM Symposium on Computational Geometry.*