Relocatable Action Models for Autonomous Navigation

Bethany R. Leffler **Michael L. Littman**
Rutgers University
Piscataway, NJ USA

A reinforcement-learning agent, in general, uses information from the environment to determine the value of its actions. Once the agent begins acting in the world, there is no further modification of its behavior by humans. This lack of human control makes the use of reinforcement learning a natural solution to the autonomous navigation and exploration problem. However, implementing algorithms from this field has not always been possible in the robotic domain.

Early work in reinforcement learning focused on approximating values of different locations by mapping them to a lower dimension function that generalized across the environment (Sutton, 1988, Tesauro, 1995). More recent work has sought to illuminate foundational issues by proving bounds on the resources needed to learn near optimal behavior (Brafman and Tennenholtz, 2002). Unfortunately, these later papers treat all locations in the world as being completely independent. As a result, learning times tend to scale badly with the size of the state space—experience gathered in one state reveals nothing about other states. The generality of these results makes them too weak for use in real-life problems in robotics and other problem domains.

The work reported in this poster builds on advances that retain the formal guarantees of recent algorithms while moving toward algorithms that generalize across states. These results rely critically on assumptions. The main assumption adopted in our current work (Leffler et al., 2007) is that states belong to a relatively small set of *types* that determine their transition behavior. In a navigation task, states can be grouped by terrain type—in areas of similar terrain, a robot's action model (how its actions affect its state) will be similar. Once correlations are discovered, not all actions have to be explored in every state to fully determine a near optimal policy. Through the use of *relocatable action models*, transition functions learned in one state can be used to calculate the results of actions in similar states. By sharing information between homogeneous states (Leffler et al., 2005), learning time is greatly reduced by limiting the number of actions that need to be performed before the world is well modeled.

Learning efficiency can be improved further by leveraging visual input. If type classification is performed visually, the robot does not have to visit a location to determine its terrain type. Visual segmentation on texture allows us to perform such classifications on terrain images.

With our proposed algorithm, the agent acquires an image of the world and classifies the states into types using texture segregation. The agent then explores all of its actions in a representative sample of states in the environment. Exploiting the assumption of terrain types mentioned above, a generalized action model for the learned states is applied to other states of similar terrain, greatly improving learning time when there are many more states than terrain types.

This work was implemented on a Lego Mindstorm wheeled robot in a 4 foot by 4 foot domain with two ground surfaces. Using our algorithm, the agent was able to obtain action models for each terrain and reach the goal on the first run. After five runs, the agent performed the learned policy continuously. This approach was compared against RMAX, which after 50 runs was not able to reliably reach the goal.

References

Brafman, R. I. and Tennenholtz, M. (2002). R-MAX—a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231.

Leffler, B. R., Littman, M. L., and Edmunds, T. (2007). Efficient reinforcement learning with relocatable action models. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*.

Leffler, B. R., Littman, M. L., Strehl, A. L., and Walsh, T. J. (2005). Efficient exploration with latent structure. In *Proceedings of Robotics: Science and Systems*.

Sutton, R. S. (1988). Learning to predict by the method of temporal differences. *Machine Learning*, 3(1):9–44.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, pages 58–67.