An Efficient Hierachy for Continuous State Multi-goal MDPs

Sean Harris, Bernhard Hengst, Maurice Pagnucco

I. Motivation

Reinforcement Learning problems, like many other robotics topics, are often high dimensionality problems that are difficult to solve efficiently. The use of hierarchy promises to alleviate the curse of dimensionality and simplify learning problems, however this hasn't been achieved yet in many robotics applications. Diettrich was able to show that discrete problems can be decomposed across a hierarchy with his MAXQ approach[1], however researchers have been unable to mirror that success for the continuous case. This work is aimed at working towards a similar hierarchical decomposition of a reinforcement learning problem except over a continuous state space rather than a discrete one.

The current hierarchical data structure being investigated is the Airports Hierarchy, which was proposed by Moore, Baird and Kaelbling in [2] and presents an algorithm to automatically generate a hierarchy for efficiently solving MDPs with multiple goals. The hierarchy allows for substantial reductions in space requirements and learning time when compared to niavely learning for multiple goals. It works well in a discretised environment but is yet to be effectively utilised in a continuous state space, which is what we are aiming to do.

Multi-goal MDPs (which this hierarchy is developed for) are applicable to many real-life robotics applications. The obvious example is a 2D navigation task where a robot learns the best way to navigate between various waypoints that may change at any point. Another application area is bipedal motion, where each goal represents a different reward function. The reward function can be varied to change the behaviour of the robot, from standing to walking to standing on one leg. Both of these applications are being investigated as part of this work.

II. Current Work

We have identified numerous tasks to tackle during the development of this project. They include:

- Utilising the data from the sub-MDP policy and applying a function approximator to act effectively in a continuous environment
- Focusing on the method of function approximation to produce a value function that smoothly transitions across borders
- Maintaining a smooth value function in challenging environments with harsh boundaries (eg walls) present

Intial work was able to successfully employ a crude function approximator to learn a policy with a continuous state space. The function approximator was simply to apply the policy of the nearest location to the current state. The result of this was a discontunuous value function that was represented roughly by a step function.

More recent work has been able to successfully employ a more sophisticated function approximator. It utilises up to four nearest neighbour states and averages the value function based on the distance to each. This has been able to provide a much smoother value function in many cases, however is less elegant when faced with more challenging cases. These challenging cases include acting near the edge of the sub-MDP, near walls and near the edge of the state space. An example of the policy and value function with this function approximator is shown below in Figure 1.

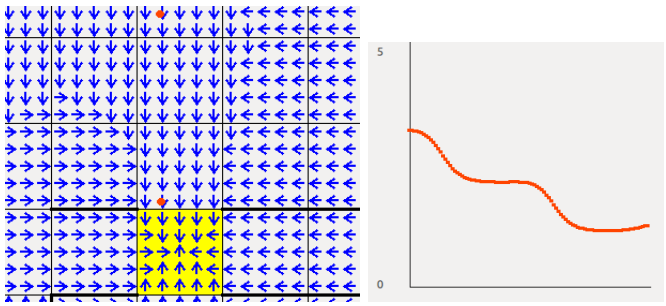

Fig. 1. A policy for the continuous airports hierarchy on a small section of maze. The yellow section is the goal region and the blue arrows represent the policy at that point. The graph shows the value function between the two orange dots

In the future we hope to continue to extend this hierarchy into the continuous state space. In particular, the focus of this research will be to obtain a smooth value function across the entire continuous state space.

References

[1] T. Dietterich. An overview of MAXQ hierarchical reinforcement learning. *Symposium on Abstraction, Reformulation, and Approximation (SARA)*, pages 26–44, 2000.

[2] Andrew W. Moore, Leemon C. Baird, and Leslie Kaelbling. Multi-value-functions: Efficient automatic action hierarchies for multiple goal mdps. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 1316–1323. Morgan Kaufmann, 1999.