Autonomous Hierarchical POMDP Planning from Low-Level Sensors

Shawn Squire and **Marie desJardins**
University of Maryland, Baltimore County
{ssquire1,mariedj}@umbc.edu

Abstract

There are currently no strong methods for planning in a stochastic domain, with low-level sensors that are limited and possibly inaccurate. Existing architectures have flaws that make their use in a real-world environment impractical. We propose an architecture that utilizes POMDPs to create a hierarchical planning system. This system is capable of developing macro-actions that can expedite planning on a large scale, and can learn new plans quickly and efficiently, without deliberate design by the programmer.

Introduction

For a robot to properly and fully operate autonomously in a real-world environment, the robot must be capable of planning for long-term goal completion, while also responding quickly and immediately to situations that put the robot in immediate damage. The agent should be able to use information learned online to adapt to changes in the world, and to learn how to operate in environments other than the one in which it was trained. For example, the agent may be placed in a domain where its existing knowledge does not include any possible solution to reach the goal. However, upon exploration, it should be able to derive a correct plan to reach the goal state.

Behaving intelligently in an unfamiliar environment is especially difficult if the agent only has access to low-level sensors that provide partial, inaccurate, and limited information about the environment. If the environment is directly inferred from the error-prone input sensor data, the agent will potentially develop an incorrect model of the world, due to noise. Additionally, planning over all possible input sensor readings leads to an intractable design, especially if the sensor readings contain continuous information. However, if the agent is able to generalize low-level information, a more stable model may be developed that is less prone to noise and irrelevant data (Andre and Russell 2002).

This proposal discusses autonomous hierarchical planning for agents with low-level sensors. Hierarchical planning allows the agent to learn a global model of the world, to adapt this model based on low-level interactions, provide feedback between lower- and upper-level thought processing, and improve the planning and learning capabilities of the agent compared to alternative architectures. Additionally, hierarchical planning enables the agent to derive macro-actions, resulting in improved performance on repeated tasks.

Background

Architectures that learn to react to immediate circumstances while simultaneously planning for long-term goals typically utilize a two-layer architecture: low-level reacting is performed actively, while high-level planning interjects actions to "steer" the agent towards the intended goal. Sutton provides a general framework for this approach (Sutton 1991; 1990), where the reaction level is handled by traditional reinforcement learning, and the high-level planning utilizes any available planning algorithm.

As demonstrated in 1998 (Kaelbling, Littman, and Cassandra 1998), a Partially Observable Markov Decision Process (POMDP) is advantageous when planning in partially observable and stochastic domains. POMDPs are robust to errors in sensor data, since the probability distribution will not be affected by slight differences in similar sensor data. POMDPs also contain the capability to learn online, despite the inaccuracy of sensor data (Shani, Brafman, and Shimony 2005).

However, reinforcement learning at a low-level is not a preferred method, largely due to a tendency to overfit (Pyeatt and Howe 1999) (especially popular approaches such as neural networks with back-propagation). Additionally, transferring the learned information across domains is nontrivial, and autonomous reinforcement learning domain transfer is still under active research (Konidaris and Barto 2006).

Hierarchical Planning

By utilizing POMDPs at all levels, the agent will be able to benefit from POMDPs at the low-level to avoid the pitfalls of standard reinforcement learning. Planning in the proposed system will be layered by complexity of tasks, with a bottom level for immediate planning that consists of

atomic actions, a toplevel that is responsible for attempting to formulate a plan to reach the goal state, and multiple mid-level POMDPs that can be dynamically created to represent macro-actions for more complex actions.

This layered approach to planning is similar to the architecture proposed by Albus in 1990 . However, Albus' architecture has a fixed number of levels whose plans are decomposed from higher levels. These levels have specific object interaction concerns and time periods to remember and plan for, so there is no dynamic scaling for tasks that are more or less complex. Finally, the architecture requires a memory at each level to accurately plan, whereas POMDPs do not require a memory, due to the Markov property.

Using a planner on the lowest level provides benefits over standard reinforcement learning. While reinforcement learning can respond immediately to a given input based purely on the input, POMDPs contain more information about what state it believes it is in, and make more informed judgements about what actions to take. Therefore, the agent can maintain a suitable belief about its state to act on, rather than reacting just on input information.

Issues with speed and tractability of POMDP at the lowest level can be resolved using approximation methods (Zhang and Liu 1997; Gordon, Roy, and Thrun 2011), which prioritize reaction time at the expense of accuracy. Additionally, fast reaction times can be obtained by using a Monte-Carlo algorithm for planning (Silver and Veness 2010). Finally, the POMDPs allow for compression (Poupart and Boutilier 2003) that will make the solution more tractable and focused on relevant details.

Mid-levels can be utilized to expedite planning on actions. Current research shows the possibility to derive macro-actions and hierarchical planning based on POMDPs (Theocharous and Mahadevan 2002; Theocharous and Kaelbling 2003). These methods enable dynamic creation of levels that operate in parallel. These mid-levels may be decomposed from higher levels by splitting a complex macro-action into more fine-grained atomic actions.

The top level is responsible for deriving the plan for the agent to reach the goal state. The plan at the highest level will guide the agent in the steps required to move from the current state to the given goal state, and is the first layer to be decomposed into subtasks at lower levels.

Conclusion

The proposed hierarchical architecture allows an agent to plan and interact completely autonomously. The POMDP framework allows the agent to operate in a stochastic and partially observable world, with potentially inaccurate sensors, but still function well. In addition to these benefits, the multiple layers allow for the system to expedite learning and reaction time, to improve performance, and to make highly complex domains more tractable.

References

Albus, J. S. 1990. A theory of intelligent systems. In *Proceedings of the 5th IEEE International Symposium on Intelligent Control 1990*, volume 15, 329–342.

Andre, D., and Russell, S. J. 2002. State abstraction for programmable reinforcement learning agents. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 119–125.

Gordon, G.; Roy, N.; and Thrun, S. 2011. Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research* 23(1):1–40.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1-2):99–134.

Konidaris, G., and Barto, A. 2006. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, 489–496.

Poupart, P., and Boutilier, C. 2003. Value-directed compression of POMDPs. In *Proceedings Neural Information Processing Systems*, volume 15, 1547–1554.

Pyeatt, L. D., and Howe, A. E. 1999. Integrating POMDP and reinforcement learning for a two layer simulated robot architecture. In *Proceedings of the Third International Conference on Autonomous Agents*, 168–174.

Shani, G.; Brafman, R.; and Shimony, S. 2005. Model-based online learning of POMDPs. In *European Conference on Machine Learning*.

Silver, D., and Veness, J. 2010. Monte-carlo planning in large POMDPs. *Advances in Neural Processing Information Systems* 23:2164–2172.

Sutton, R. S. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Proceedings of the Seventh International Conference on Machine Learning* 216:216–224.

Sutton, R. S. 1991. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin* 2(4):160–163.

Theocharous, G., and Kaelbling, L. P. 2003. Approximate planning in POMDPs with macro-actions. *Advances in Neural Processing Information Systems* 17.

Theocharous, G., and Mahadevan, S. 2002. Approximate planning with hierarchical partially observable markov decision process models for robot navigation. *Robotics and Automation* 2:1347–1352.

Zhang, N. L., and Liu, W. 1997. A model approximation scheme for planning in partially observable stochastic domains. *Journal of Artificial Intelligence Research* 7:199–230.