

A Weighted Instance-Based Algorithm for Situated Autonomous Robot Learning

Carlos H. C. Ribeiro and Elder M. Hemerly

Instituto Tecnológico de Aeronáutica
Divisão de Engenharia Eletrônica
Praça Mal. Eduardo Gomes, 50
12228-900 São José dos Campos - SP, Brazil
c.ribeiro@ieee.org hemerly@ele.ita.cta.br

Abstract

We report preliminary results on a weighted instance-based algorithm for the problem of autonomous robot learning. The algorithm combines the K -nearest neighbour technique and a distance metric which provides selective spreading of learning updates on the experience space, with the aim of minimizing the problem of partial state observability produced by local sensor readings and insufficient global information. Results show that the algorithm generated a good action policy for a simulated guidance robot with two basic behaviours (preprogrammed obstacle avoidance and learned target approximation).

Introduction

The problem of autonomous robot learning is one of the foremost research subjects in Artificial Intelligence today. The theoretical solidity of some recent results on Reinforcement Learning (RL) algorithms and the advanced state of research on architectures that combine multiple behaviours have brought a fresh interest in the development of methods for model-free robot navigation and guidance. However, there are many open issues to be tackled. First, plain application of autonomous learning techniques on functionally monolithic robots is unlikely to yield interesting results, due to theoretical restrictions on the cost functions that can be minimized. Second, the problem of partial observability (incomplete state description) destroys an important requirement for the optimal operation of RL techniques, namely the Markovian assumption (Chrisman 1992). Third, the size of the state spaces normally found in practical applications is much larger than those of the 'toy-problems' normally used to demonstrate validity of techniques, and this may force application of approximation techniques which can produce divergence (Boyan & Moore 1995).

In this paper, we report on a learning algorithm that, embedded in a particular architecture, can tackle those problems. The algorithm aims at eliminating the partial observability problem by producing action value updates (in the spirit of the Q-learning algorithm (Watkins 1989)) that consider the history of observation instances instead of just current observations,

in what can be seen as an attempt at transferring the concept of information vectors and sufficient statistics (Striebel 1965) to the standard RL approach. The basic idea is to consider similarity between past histories as the underlying concept defining if two given observations correspond to the same environmental state, whilst at the same time using the distances calculated by the similarity metric to define how strongly a history of instances influence a given observation action value. This method is part of the family of *instance-based* methods, that instead of predefining an architecture proportional to the size of the state space, record only these parts of the state that are actually visited, thus avoiding combinatorial explosion derived from multiple sensor readings. Finally, we avoid slow convergence by using an architecture that uses the algorithm only for learning a *target approximation* behaviour, combined with a built-in reflex behaviour for avoiding collisions. We also consider a source of coarse global information on the structure of the state space.

Weighted Nearest-Neighbour Matching

The idea of using *instance-based* methods for model-free learning of action policies consists on the time-ordered storage of instances (experiences) undergone by the learning agent. Each experience is a tuple (a_t, o_t, r_t) , composed by the chosen action a_t and consequents observation o_t and reward r_t .

McCallum (McCallum 1996b) proposed calculating action values for each visited instance by averaging expected future rewards associated with the K nearest experiences of the instance sequence, in an analogy with the K -nearest neighbour method for pattern recognition in geometric spaces. We propose here a *Weighted Nearest-Neighbour Matching* technique, which produces updates whose intensity is proportional not only to the number of neighbouring observations, but also to how close (history wise) those are from the current instance. Moreover, the current reinforcement is used, instead of the formerly proposed average over reinforcements received by the neighbouring experiences when they were experimented.

Formally, the Weighted Nearest-Neighbour Matching (WNNM) algorithm is as follows. For a given experi-

ence $s_t = \langle a_t, o_t, r_t \rangle$ at time t , do

1. Find the K -nearest neighbours of this instance in the time-ordered chain, using a 'history string match', i.e., by selecting the K former instances that have a past history more similar to that of the current instance. Store the distances $d(s_i, s_t)$ between those instances and the current instance in an auxiliary vector (of size K). The similarity metric is defined recursively as:

$$d(s_i, s_t) = \begin{cases} 1.0 + 0.5n + d(s_{i-1}, s_{t-1}), & \text{if } a_{i-1} = a_{t-1} \wedge o_{i-1} = o_{t-1} \\ 0.0, & \text{otherwise} \end{cases}$$

where n is the number of identical sensor readings. Note that similarity between global readings has twice the weight of similarity between local sensor readings.

2. The action value for each possible action a is computed as a weighted average of the corresponding stored action values of the K -nearest neighbours, that is,

$$Q(s_t, a) = \frac{\sum_{i=1}^K d(s_i, s_t) Q(s_i, a)}{\sum_{i=1}^K d(s_i, s_t)} \quad (1)$$

3. Update the action values of each neighboring instance according to

$$\Delta Q(s^i, a_i) = \alpha d(s_i, s_t) (r_t + \gamma \max_a Q(s_t, a) - Q(s_i, a_t)) \quad (2)$$

where α is the learning rate and γ is a temporal discount factor.

4. Select the next optimal action to be taken as $a_{t+1} = \arg \max_a Q(s_t, a)$ and apply it (possibly with some added noise to improve exploration) to the learning agent. Record the resulting instance $s_{t+1} = \langle a_{t+1}, o_{t+1}, r_{t+1} \rangle$.
5. Repeat steps above until performance produced by action policy reaches acceptable level.

Equation 2 corresponds to a modification of the standard Q-learning update

$$\Delta Q(x_t, a_t) = \alpha (r_t + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t)) \quad (3)$$

with states x replaced by instances s and with an additional 'spreading' factor $d(s_i, s_t)$ responsible for multiple weighted updates on the instance-action space. It is a well-known fact that Q-learning converges (under standard stochastic algorithms requirements) to optimality with respect to $\lim_{M \rightarrow \infty} E[\sum_{t=0}^M \gamma^t r_t]$ (i.e., $\arg \max_a Q(x, a)$ is an optimal action with respect to maximization of the expected sum of discounted reinforcements) (Watkins & Dayan 1992).

Experimental Results

We tested WNNM on a model-free learning guidance task for a situated robot. This involves obtaining an optimal stochastic policy in a very large state space, under weak structural assumptions.

Methodology

Our testbed is a simulated Khepera robot (Michel 1996), acting on a simple room with some obstacles and a light source. The goal of the robot is to reach a position that maximizes average light reception, whilst at the same time avoiding hitting obstacles and walls (Figure 1).

The robot can perform three actions: go forward, turn right and turn left. The first moves the robot a distance corresponding to half the diameter of its physical base. The right and left turning actions correspond to ninety degrees turns, eastbound and westbound, respectively. Fine tuning control of these turns is provided by a compass attached to the robot that can measure its relative orientation with respect to a fixed Cartesian system.

A second source of global information is available to the robot. We divide the environment into six rectangular grids, and the robot can always assess in which of these grids it is. This information helps discriminating states, but it is not sufficient for learning guidance paths as they do not ensure first-order Markov observations. In addition to this restricted global information, the robot has eight sensors arranged around

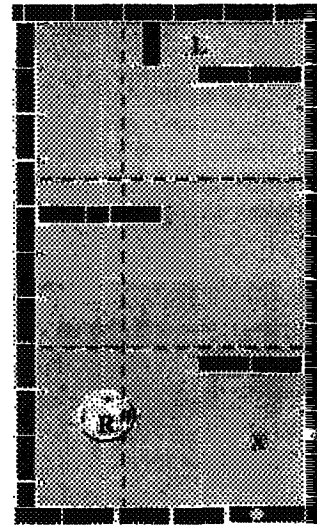


Figure 1: The simulated robot (R) and the environment where it acts. The target behaviour is to get from an initial position (X) to a position as close as possible to the light source (L), whilst at the same time avoiding hitting the obstacles. The global information about orientation correspond to the six rectangular regions marked by the discontinuous lines.

it, each of which providing information about both light and obstacle presence. We used only four of those sensors (the front ones), with a 4 levels discretisation. Combined with the global information about position and orientation, this corresponds to a space of $6 \times 4 \times 4^4 \times 4^4 = 1,572,864$ states. Many of these, however, are never visited by the robot.

As we plan in the near future to test the algorithm in a real robot, we considered a reflex obstacle-avoidance behaviour independently implemented, called upon only when obstacle hitting is upcoming. This behaviour produces a strong negative reinforcement, thus making learning *to avoid its use* (instead of learning to avoid hitting walls) a simpler requirement for the robot. Embedding reflex rules in this way guarantees that the robot is operational from the very beginning of the learning process (del R. Millán 1996).

Results

Figure 2 shows the path followed by the robot for the first 100 operation steps, for three different initial positions, after 10,000 training steps carried out under an explorative action policy ($P(\text{random action})=0.1$). Notice that the trained robot reaches a position close to the light source, but not as close as to hit the walls. Once it reaches this position, it produces consecutive turn left-turn right actions, in an attempt to keep it oriented towards the light source.

Note that the learned paths present deviations from what would be the optimal ones. This phenomenon was also observed by McCallum (McCallum 1996a) for an agent trying to learn the spaceship docking task under a K -nearest neighbour weightless instance-based technique. He observed that his agent did learn an optimal path, apart from some unnecessary superstitious turns ('dances'), caused by the agent difficulty in differentiating environmental noise from state space structure. He also observed that better paths (without the 'dance') were reached after a much longer number of training steps. We believe that this would also happen in our

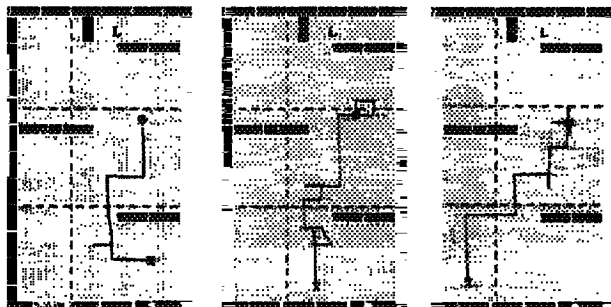


Figure 2: Paths (100 steps) learned by the robot after training. The initial position is marked with an X. Note the presence of superstitious behaviour ('dance'), characterised by unnecessary turns before the final position is reached.

experiment, even though the number of steps necessary to avoid 'dancing' may be extremely high due to the much increased complexity and state space size of the problem compared to the spaceship docking task. An alternative solution to this problem is the incorporation of advice from an external teacher at advanced stages of learning (Maclin & Shavlik 1994).

Although the use of a weighted metric might be unnecessary if a feature extraction mechanism can generate percepts that compensate for the intolerance to noise of a 'hard' metric (Mataric 1990), we did notice an improvement over NSM for the reported task, indicating that the use of a weighted metric may still be important when the domain knowledge is insufficient to account for noisy percepts.

Related Literature

Reinforcement Learning has been well studied and demonstrated in the perfectly observable, small state-space case (Watkins 1989; Sutton & Barto 1998). In particular, the use of a 'spreading' factor for Q-learning based on a distance metric was studied in (Ribeiro 1998). In more realistic, partially observable processes, instance-based methods have been proposed by some authors (McCallum 1996a; Moore 1992; Schneider 1995), usually in very simple problems made extremely complex by the partial observability condition. Alternative techniques include the use of recurrent networks for past history encoding (Whitehead & Lin 1995), and attempts at on-line state estimation based on Hidden Markov Models concepts (Chrisman 1992).

Conclusion

We reported on an instance-based method for autonomous learning under local perception constraints and coarse global information. The method can reduce the perceptual aliasing problem by producing updates on an instance chain that are proportional to the distance between past histories of instances. The results showed that the algorithm generated a good action policy for a simulated guidance robot with two basic behaviours: preprogrammed obstacle avoidance (whose inhibition is part of the learning process) and learned target approximation.

Our future developments will include setting up experiments on the proposed algorithm and architecture for a real robot and making a more comprehensive comparative analysis among it and other instance-based algorithms.

Acknowledgements

We are grateful to FAPESP (grant 97/07630-5) and CNPq (grant 300158/95-5 RN) for their financial support.

References

Boyan, J. A., and Moore, A. W. 1995. Generalization in reinforcement learning: Safely approximating

the value function. In Tesauro, G.; Touretzky, D. S.; and Leen, T. K., eds., *Advances in Neural Information Processing Systems 7*. MIT Press.

Chrisman, L. 1992. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Procs. of the 10th National Conf. on Artificial Intelligence*, 183–188.

del R. Millán, J. 1996. Rapid, safe and incremental learning of navigation strategies. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* 26(3):408–420.

Maclin, R., and Shavlik, J. W. 1994. Incorporating advice into agents that learn from reinforcements. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 694–699.

Mataric, M. J. 1990. A distributed model for mobile robot environment learning and navigation. Master's thesis, Massachusetts Institute of Technology.

McCallum, A. K. 1996a. *Reinforcement Learning with Selective Perception and Hidden State*. Ph.D. Dissertation, University of Rochester.

McCallum, R. A. 1996b. Hidden state and reinforcement learning with instance-based state identification. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics* 26(3):464–473.

Michel, O. 1996. *Khepera Simulator Package version 2.0*. University of Nice Sophia-Antipolis. Downloadable from WWW address <http://wwwi3s.unice.fr/~om/khep-sim.html>.

Moore, A. W. 1992. *Efficient Memory-based Learning for Robot Control*. Ph.D. Dissertation, University of Cambridge.

Ribeiro, C. H. C. 1998. Embedding a priori knowledge in reinforcement learning. *Journal of Intelligent and Robotic Systems* 21(1):51–71.

Schneider, J. G. 1995. *Robot Skill Learning Through Intelligent Experimentation*. Ph.D. Dissertation, University of Rochester.

Striebel, C. T. 1965. Sufficient statistics in the optimal control of stochastic systems. *Journal of Math. Analysis and Applications* 12:576–592.

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

Watkins, C. J. C. H., and Dayan, P. 1992. Q-learning. *Machine Learning* 8(3/4):279–292.

Watkins, C. J. C. H. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation, University of Cambridge.

Whitehead, S. D., and Lin, L.-J. 1995. Reinforcement learning of non-markov decision processes. *Artificial Intelligence* 73:271–306.