Integrating Experimentation and Guidance in Relational Reinforcement Learning (Extended Abstract)

Kurt Driessens [a] Sašo Džeroski [b]

[a] Department of Computer Science K.U.Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium
kurt.driessens@cs.kuleuven.ac.be
[b] Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
saso.dzeroski@ijs.si

The full paper on this topic appears in the Proceedings of the Nineteenth International Conference on Machine Learning. [1]

Q-learning [3] is a form of reinforcement learning where the optimal policy is learned implicitly in the form of a Q-function, which takes a state-action pair as input and outputs the quality of the action in that state. The optimal action in a given state is the action with the greatest Q-value. When dealing with large state spaces Q-learning encounters two major problems.

The first limitations of standard Q-learning is related to the number of different state-action pairs that may exist. The Q-function can in principle be represented as a table with one entry for each state-action pair. When states and actions are characterised by parameters, the number of such pairs grows combinatorially in the number of parameters and thus can easily become very large, making it infeasible to represent the Q-function in tabular form, let alone learn it accurately (convergence of the Q-function only happens after each state-action pair has been visited many times). This problem is typically solved by integrating into the Q-learning algorithm an inductive learner, which learns a function that generalises over given state-action pairs. Thus reasonable estimates of the Q-value of a state-action pair can be made without ever having visited it. Examples include neural networks, nearest neighbour methods and regression trees.

A relational learner is employed in "relational reinforcement learning" or RRL. RRL uses first order representations for states and actions, and learns a first order regression tree that maps these structural descriptions onto real numbers. The use of first order representations gives RRL a broader application domain than classical Q-learning approaches. Examples of such relatively complex applications described in more detail in this paper, include learning to solve simple planning tasks in a blocks world, or learning to play certain computer games (Digger, Tetris).

In structural domains, the state space is typically very large, and while a relational learner can provide the right level of abstraction to learn in such a domain, there remains the problem that rewards may be distributed very sparsely in this state space. Using random exploration through the search space, rewards may simply never be encountered. In some of the application domains mentioned above this prohibits RRL from finding a good solution.

While plenty of exploration strategies exist [4], few deal with the problems of exploration at the start of the learning process. It is exactly this problem that we are faced with in our RRL setting. There is, however, an approach which has been followed with success, and which consists of guiding the Q-learner with examples of "reasonable" strategies, provided by a teacher [2]. Thus a mix between the classical unsupervised Q-learning and (supervised) behavioural cloning is obtained.

It is the suitability of this approach in the context of RRL that we explore in this paper. We discuss how guidance can be incorporated in a Q-learning approach, and how this was done in our RRL algorithm. We demonstrate the feasibility of this approach in three domains, the blocks world and two computer games (Digger and Tetris). Three different forms of guidance are considered: traces (action sequences) generated by hand-coded policies, traces generated by policies learned by RRL and traces of a human performing the task at hand. In all three cases, the use of guidance followed by experimentation improves performance over using experimentation only, either in terms of the overall performance level achieved or the convergence speed.

We also observe that one has to be careful about supplying the learning algorithm with too much "perfect guidance" right at the start. Coupled with the use of a generalisation engine, providing optimal actions only does not allow to learn to distinguish between optimal and non-optimal actions. Good guidance will show the learning algorithm both optimal and non-optimal actions in a great variety of states. Both restricting the visited states during guidance and limiting the guidance policy to take only correct actions will have a negative influence on the effectiveness of the offered guidance. The variety in the visited states is probably the hardest to achieve when constructing guidance traces.

References

[1] K. Driessens and S. Džeroski. Integrating experimentation and guidance in relational reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*, pages 115–122, 2002.

[2] W. D. Smart and L. P. Kaelbling. Practical reinforcement learning in continuous spaces. In *Proceedings of the 17th International Conference on Machine Learning*, pages 903–910. Morgan Kaufmann, 2000.

[3] C. Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge., 1989.

[4] M. Wiering. *Explorations in Efficient Reinforcement Learning.* PhD thesis, University of Amsterdam, 1999.