Knowledge Revision for Reinforcement Learning with Abstract MDPs

(Extended Abstract)

Kyriakos Efthymiadis
The University of York
ke517@york.ac.uk

Sam Devlin
The University of York
sam.devlin@york.ac.uk

Daniel Kudenko
The University of York
daniel.kudenko@york.ac.uk

ABSTRACT

Reward shaping has been shown to significantly improve an agent's performance in reinforcement learning. As attention is shifting from tabula-rasa approaches to methods where some heuristic domain knowledge can be given to agents, an important problem that arises is how can agents deal with erroneous knowledge and what is the impact to their behavior both in a single- as well as a multi-agent setting where agents are faced with conflicting goals. Previous research demonstrated the use of plan-based reward shaping with knowledge revision in a single agent scenario where agents showed that they can quickly identify and revise erroneous knowledge and thus benefit from more accurate plans. Moving to a multi-agent setting the use of individual plans as a source of reward shaping has not been as successful due to the agents' conflicting goals. In this paper we present the use of MDPs as a method to provide heuristic knowledge coupled with a revision algorithm to manage the cases where the provided domain knowledge is wrong. We show how agents can deal with erroneous knowledge in the single agent case and how this method can be used in a multi-agent environment for conflict resolution.

Categories and Subject Descriptors

Computing methodologies [**Artificial Intelligence**]: Learning

General Terms

Experimentation

Keywords

reinforcement learning, reward shaping, knowledge revision

1. INTRODUCTION

In earlier work on *knowledge-based reinforcement learning* [2, 3] it was demonstrated that the incorporation of domain knowledge in reinforcement learning via reward shaping can significantly improve the speed of convergnce. However problems arise when the expert knowledge provided is erroneous [1].

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

In this paper we propose a method to revise knowledge by the use of abstract MDPs [4]. We compare our revision method to that using plan-based reward shaping [1] and we demonstrate empirically that our agent can achieve similar performance in the single-agent case. In the multi-agent setting, we are interested in those cases where communication and goal sharing is not allowed and agents receive decentralised shaping. We demonstrate that the agents using abstract MDP reward shaping manage to efficiently coordinate to learn a much better policy compared to the agents using plan-based reward shaping when both receive decentralised shaping.

2. EVALUATION DOMAIN

We evaluate our method on the flag-collection domain. An agent is modelled at a starting position from where it must move to the goal position. The agent needs to collect flags which are spread throughout the maze. At each time step, the agent is given its current location and the flags it has already collected. From this it must decide to move up, down, left or right and will complete its move provided it does not collide with a wall. The scenario ends when the agent reaches the goal position. This domain is also used as a multi-agent setting by adding a second agent.

3. THE REVISION PROCESS

We allow the abstract MDP agent to constantly update its transition probabilities according to its experiences in the low level environment. If the agent experiences a state transition which is not present in the abstract MDP, it is added to the current set of transitions. The MDP is solved and the new value function is used for shaping. This results in states which are not experienced, either because of wrong domain knowledge or because of the environment dynamics, to assume a low value and consequently will not be used for shaping due to their low potential.

4. EVALUATION

Single-Agent Revision

In the case of incorrect knowledge, the agents are provided with knowledge which contains extra information which is not present in the environment. In the incomplete knowledge case, the agents are provided with knowledge which is missing important goals. The results for the incorrect case are shown in Figure 1 and the incomplete in Figure 2.

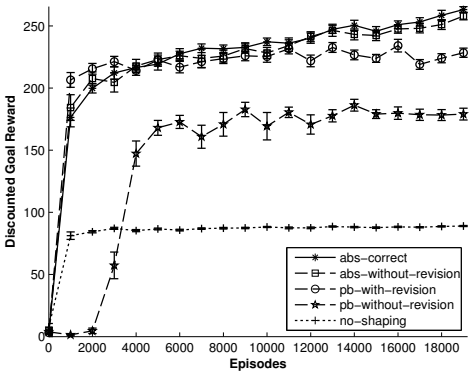

Figure 1: Incorrect knowledge comparison.

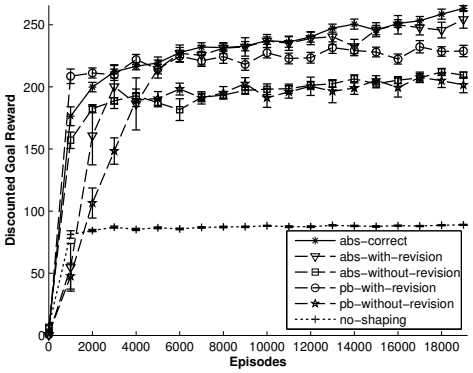

Figure 2: Incomplete knowledge comparison.

While the plan-based agent is impacted and a need for revision is apparent, this is not the case for the abstract MDP agent in the incorrect case. Since a value function is used as a reward shaping source, every state the agent finds itself in will have a potential that will lead to the goal. These multiple paths to the goal mean that the agent will never be left without guidance.

The plan-based agent however receives only a single path to the goal. Therefore if the agent cannot achieve a step in the provided plan because the path does not exist, it does not receive any further guidance after that point.

In the incomplete knowledge case, our agent manages to quickly identify the parts which are missing from the provided MDP. By adding these new transitions it encounters in the low-level to the abstract MDP, as described in Section 3, it manages to solve a more accurate MDP and thus benefit from better shaping.

Multi-Agent Results

We are interested in those cases where information sharing is not allowed and the agents are agnostic to other learning entities in the environment. To set an upper bound on performance, we also include a setting in which agents receive plan-based reward shaping from a joint-plan generated by a centralised agent. Figure 3 show the performance of the agents in the flag collection domain. The plan-based agent receiving individual shaping fails to reach a satisfying performance. The agents fail to coordinate and one of them opts out and heads to the goal location, while the other agent

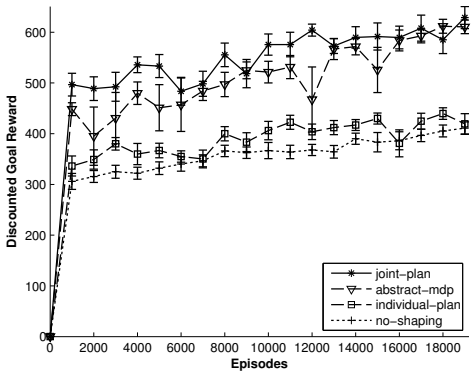

Figure 3: Multi-agent flag-collection comparison.

collects all the flags. Certain goals in the plan cannot be satisfied and as a result only one agent is able to collect all the extra rewards and learn a better policy.

The agents receiving abstract MDP shaping manage to achieve a performance similar to the agent receiving centralised shaping. When agents receive individual abstract MDP shaping in a multi-agent setting, it becomes a version of incorrect knowledge in the single agent case. As it has been shown this does not have any impact in the agent's performance and both agents are able to learn a much better policy than the individual learners using plan-based shaping.

5. CONCLUSION

We demonstrated empirically that our approach can achieve a similar behavior to the revision method using plan-based reward shaping.

We have shown that the abstract MDP method is not impacted at all when the domain knowledge is incorrect. The multiple paths leading to the goals results in the agent being able to ignore those parts of the knowledge which are incorrect.

In the incomplete knowledge case, we have shown that our revision method can update the abstract MDP transition probabilities which results in better, more accurate shaping.

We have demonstrated that the abstract MDP agents can learn to cooperate effectively in a multi-agent setting and eliminate conflicting goals when provided with decentralised shaping i.e. knowledge that ignores the presence of other agents.

6. REFERENCES

[1] K. Efthymiadis, S. Devlin, and D. Kudenko. Overcoming erroneous domain knowledge in plan-based reward shaping. In *Autonomous agents and multi-agent systems*, pages 1245–1246. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[2] K. Efthymiadis and D. Kudenko. Using plan-based reward shaping to learn strategies in starcraft: Broodwar. In *Computational Intelligence and Games (CIG)*. IEEE, 2013.

[3] M. Grześ and D. Kudenko. Plan-based reward shaping for reinforcement learning.

[4] B. Marthi. Automatic shaping and decomposition of reward functions. In *International Conference on Machine learning*, page 608. ACM, 2007.