Solving Delayed Coordination Problems in MAS

(Extended Abstract)

Yann-Michaël De Hauwere
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussel, BELGIUM
ydehauwe@vub.ac.be

Peter Vrancx
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussel, BELGIUM
pvrancx@vub.ac.be

Ann Nowé
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussel, BELGIUM
anowe@vub.ac.be

ABSTRACT

Recent research has demonstrated that considering local interactions among agents in specific parts of the state space, is a successful way of simplifying the multi-agent learning process. By taking into account other agents only when a conflict is possible, an agent can significantly reduce the state-action space in which it learns. Current approaches, however, consider only the immediate rewards for detecting conflicts. This restriction is not suitable for realistic systems, where rewards can be delayed and often conflicts between agents become apparent only several time-steps after an action has been taken.

In this paper, we contribute a reinforcement learning algorithm that learns where a strategic interaction among agents is needed, several time-steps before the conflict is reflected by the (immediate) reward signal.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms

Keywords

Reinforcement learning, coordination problems, multi-agent learning

1. INTRODUCTION

Reinforcement Learning (RL) is an unsupervised learning technique which allows agents to learn policies in initially unknown, possibly stochastic, environments, steered by a scalar reward signal they receive from the environment. This signal can be delayed, such that agents only see the effect of a certain action, several timesteps after the action was performed. Using an appropriate backup diagram which backpropagates these rewards still ensures convergence to the optimal policy [4]. When multiple agents are present in the environment, these guarantees no longer hold, since the agents now experience a non-stationary environment due to the influence of other agents [5].

Most multi-agent learning approaches alleviate the problem by providing the agents with sufficient information about each other. Generally this information means the state information and selected actions of all the other agents. As such, the state-action space becomes exponential in the number of agents.

Recent research has illustrated that it is possible to identify in which situations this extra state information is necessary to obtain good policies [3, 1] or in which states agents have to explicitely coordinate their actions [2]. These techniques rely on sparse interactions with other agents and only use the state information of the other agents if this is needed. In all these techniques however, it is assumed that the need for coordination is reflected in the immediate reward signal. However, in RL-systems a delayed reward signal is common. Similar, in a multi-agent environment the effect of the joint action of the agent is often only visible several time steps in the future.

In this paper we describe an algorithm which will determine the influence of other agents on the total reward until termination of the learning episode. By means of statistical test on this information it is possible to determine when the agent should take other agents into consideration even though this is not yet reflected by the immediate reward signal. By augmenting the state information of the agents in these situations to include the (local) state of the other agents, agents can coordinate without always having to learn in the entire joint-state joint-action space.

2. DELAYED COORDINATION PROBLEMS

The main idea behind our approach is to port the principle of delayed rewards to the framework of sparse interactions. If we think about mobile robots navigating in an environment, it is possible that there are some bottleneck areas, such as small alleys where robots will only see the fact that they had to coordinate when it is already too late, i.e. both robots are already in the alley. A similar situation in which coordination must occur is when the order in which agents enter the goal is important for the reward they can earn.

2.1 FCQ-learning

The technique we describe here uses the same basic prin-

Cite as: Solving Delayed Coordination Problems in MAS (Extended Abstract), Y-M. De Hauwere, P. Vrancx and A. Nowé, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1115-1116.

ciples as CQ-learning [1], but has been adapted to be able to deal with future coordination problems. This is why we call this approach FCQ-learning, which stands for *Future Coordinating Q-learning*. As for CQ-learning, the idea is that agents learn in which of their local states they will augment there state information to incorporate the information of other agents and use a more global system state.

The most important challenge to achieve this, is detecting in which states, the state information must be augmented. FCQ-learning makes use of a Kolmogorov-Smirnov test for goodness of fit to trigger an initial sampling phase. This statistical test can determine the significance of the difference between a given population of samples and a specified distribution. We assume the agents have converged to the correct single agent Q-values. FCQ-learning will compare the evolution of the Q-values when multiple agents are present to the values it learned when acting alone in the environment.

If a change is detected in the Q-values of a state of an agent, it will start observing the local state information of the other agents and start sampling the rewards it collects, starting from that local state until termination of the episode. Using these samples, the agent can perform a Friedmann statistical test which can identify the significance of the difference between the different local states of the other agents for its own local state. This principle is represented in Figure 1. Agent 1 starts sampling the rewards until termination of the episode in local state x^i based on the local state information y^i,y^j and y^k of Agent 2. If a significant difference is detected, the state information for x^i is augmented with the state information of agent 2 that caused this change

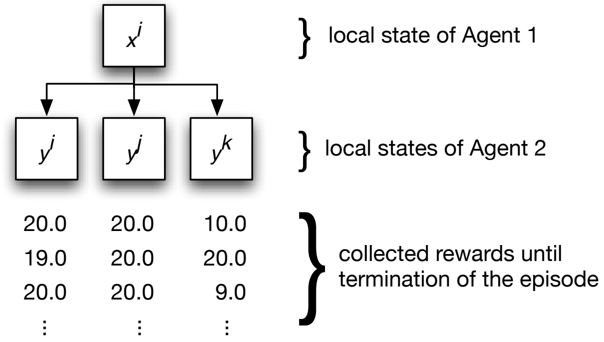

Figure 1: Agent 1 in local state x^i is collecting rewards until termination of the episode based on the local state information of agent 2.

The action selection works as follows. The agent will check if its current local state is a state which has been augmented to include the state information of other agents. If so, it will check if it is actually in the augmented state. This means that it will observe the global state to determine if it contains its augmented state. If this is the case, it will condition its action based on this augmented state information, otherwise it can act independently using only its own local state information.

If an agent is in a state in which it used the global state information to select an action it will update its joint Q-values and bootstrap using the single agent Q-values. In all other situations the normal Q-learning update rule is used.

For every augmented state a confidence value is maintained which indicates how certain the algorithm is that this is indeed a state in which coordination might be beneficial. This value is updated at every visit of the local state.

2.2 FCQ-learning with uninitialised agents

Having initialised agents beforehand who have learned the correct Q-values to complete their task is an ideal situation, since agents can transfer the knowledge they learned in a single agent setting to a multi-agent setting, adapting only their policy when they have to. This is of course not always possible. This is why we propose a simple variant of FCQ-learning. By collecting samples for every state-action pair at every timestep these single agent Q-values and the KS-test are no longer required. Despite this relaxation in the requirements for the algorithm, this results in a lot more data to run statistical tests on, most of which will be irrelevant.

3. CONCLUSION

In this paper we presented an algorithm that learns in which states of the state space an agent needs to include knowledge or state information about other agents in order to avoid coordination problems that might occur in the future. By means of statistical tests on the obtained rewards and the local state information of other agents, FCQ-learning is capable of leaning in which states it has to augment its state information in order to select actions using this augmented state information. We have described two variants on this algorithm that have a different computational complexity in terms of processing power and memory usage, due to the number of samples collected and on which statistical tests have to be performed.

Future research will focus on exploring different coordination techniques than merely selecting actions using more state information, as well as applying FCQ-learning to more complex multi-agent environments such as robosoccer. In such an application, FCQ-learning can be used to adapt strategies, based on the actions of the opponent team.

4. REFERENCES

[1] Y.-M. De Hauwere, P. Vrancx, and A. Nowé. Learning multi-agent state space representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 715–722, Toronto, Canada, 2010.

[2] J. Kok, P. 't Hoen, B. Bakker, and N. Vlassis. Utile coordination: Learning interdependencies among cooperative agents. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG05)*, pages 29–36, 2005.

[3] F. Melo and M. Veloso. Learning of coordination: Exploiting sparse interactions in multiagent systems. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 773–780, 2009.

[4] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[5] J. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Journal of Machine Learning*, 16(3):185–202, 1994.