Combining Independent and Joint Learning: a Negotiation based Approach

(Extended Abstract)

Reinaldo A. C. Bianchi
Centro Universitario FEI
São Bernardo do Campo, Brazil.
rbianchi@fei.edu.br

Ana L. C. Bazzan
Instituto de Informática / PPGC
UFRGS, Brazil.
bazzan@inf.ufrgs.br

ABSTRACT

This work presents a new class of multiagent reinforcement learning algorithms that takes advantage of negotiation in order to improve the process of action selection. In this class of algorithms, agents use communication to cooperate and negotiate over the joint actions, thus enhancing the process of action selection. In this paper a new algorithm in this class is proposed: the Negotiation-based Q-Learning (NQL), which uses negotiation in the context of the Q-Learning algorithm. Results show that allowing negotiation between agents significantly enhances the performance of the multiagent learning process.

Categories and Subject Descriptors

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

General Terms

Algorithms, Theory

Keywords

Multiagent Reinforcement Learning, Negotiation

1. INTRODUCTION

In their work, Claus and Boutilier [1] have defined two forms of multiagent reinforcement learning (MARL): Independent learners (ILs), which apply Q-learning in the classic sense, ignoring the existence of other agents, and the Joint Action Learners (JALs) that, in contrast, learn the value of their own actions in conjunction with those of other agents.

The main problem of the JAL algorithm is the size of the representation of joint actions and states, which is a key factor that limits the use of algorithms for MARL in complex problems. Another known issue of the JAL is that it is not guaranteed that the chosen set of actions is coordinated with those of other agents. This may in turn lead to agents converging to different targets. Even in cooperative games, two agents could end up with two different, possibly uncoordinated (and hence inefficient), policies.

To cope with these problems, this work presents a new class of MARL algorithms that uses negotiation to choose the actions agents execute. Negotiation is employed by independent learning agents to implement cooperative actions when it is better than to act individually. Since a centralized solution is usually not feasible in large state-action spaces, decomposing the problem into subproblems using cooperation between independent agents in some parts of the environment is a way to reduce the complexity of the problem.

2. COMBINING NEGOTIATION AND MULTIAGENT RL

To describe the class of algorithms that can be implemented by extending any MARL algorithm using negotiation, we propose a meta algorithm that is a high level description of how negotiation should be used in MARL, and serves as a template for extending traditional algorithms.

The main characteristic of the meta algorithm is that, before selecting what actions to perform, agents (when acting independently) negotiate with the aim of deciding which actions to take. This is shown in Algorithm 1, where s_i is the state that describes the system at a defined moment, as seen by agent i, and $\mathcal{A}$ is the set of actions to be used.

The negotiation algorithm used in this work is based on the one proposed by Fabregues and Sierra [2], which consists of "repeat a sequence of: a number of negotiation rounds up to the time limit, a selection of actions from the set of agreed upon joint plans and their execution. When new messages arrive, the algorithm check if it is a proposal. If it is, the message is stored in the set of proposals. This set is

Algorithm 1 The Negotiation MARL Meta-algorithm

```
Initialise $\hat{Q}_i$ arbitrarily.
repeat
    Observe the state $s_i$.
    Negotiate with other agents the actions $A$ to be used.
    Execute action $a_i$.
    Receive the reinforcement $r_i$.
    Observe the next state $s'_i$.
    Update the values of $\hat{Q}_i$.
    $s_i \leftarrow s'_i$.
until some stopping criteria is reached.
```

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

periodically checked to select a subset of proposals that can be jointly acceptable. The rest of proposals are rejected. If none of the proposals is good enough, a new deal is selected and the negotiation round is finished. Every proposal is stored until an answer is received or a timeout fires" [2].

We use their algorithm for the purpose of negotiating in a learning scenario as follows: an agent i uses an $\epsilon - greedy$ strategy to choose a set of actions $\vec{a}$. This set may contain actions for all agents, for some of the agents, or only for agent i itself. We remark that the latter occurs especially in the beginning of the learning process. If $\vec{a}$ involves two or more agents, i formulates a proposal regarding a joint action and sends this to other agents. This proposal also contains the expected utility of taking the joint action. During a certain period of time (set by a variable called "patience"), the agents collect proposals from the others.

Following this phase, i.e., after all agents have collected a set of proposal, each agent chooses the proposal that maximizes its expected return, and informs others which action was selected. It may occur that agent i decides to act as an IL, because the individual action a_i has the best utility for this agent when compared to those that were proposed. In this case, i must at least inform other agents of its action selection, so that the others can update their Q-values using the correct action for every agent.

It is important to notice that if the action the agent chooses to execute was a random exploration move, then the agent will not negotiate. It only informs other agents about this action. This is a characteristic that enables existing convergence proofs to hold for this new class of algorithms.

3. VALIDATION OF THE ALGORITHM

To validate the mentioned new class of MARL algorithms, the algorithm called Negotiation based Q-Learning (NQL) is proposed, which uses negotiation in the well-known RL algorithm Q-Learning. By means of negotiation, NQL can implement cooperative actions only when it is better than to act individually. Therefore, it can be seen as an IL that acts as a JAL in some situations.

Empirical evaluations were carried out in a simulator for the robot soccer domain that extends the one proposed by Littman in [3]. In this domain two teams, A and B, with 2 players each, compete in a 4×5 grid (agents are cooperative inside one team, competitive between the teams). The allowed actions are: move (north, south, east and west) or pass the ball to another agent. The action "pass the ball" from agent a_i to a_j is successful if there is no opponent between them. If there is an opponent, it will catch the ball and the action will fail. A complete description of this domain can be found in [3].

In the first experiment, two teams using the algorithms Q-learning and NQL play against an opponent team in which agents move randomly. Thirty training sessions were run for each team, with each session consisting of 5000 games of 10 trials. A trial finishes whenever a goal is scored by any of the agents. The parameters used in the experiments are identical to those used by Littman [3]).

Figure 1(top) shows the learning curves for the algorithms, presenting the average goal difference in each game (i.e., the goals scored by the learning team minus the goals scored by the opponent - in this case, the random team). It is possible to verify that the Q-Learning is outperformed by the NQL at the initial learning phase, and that as the games proceed, the performance of both algorithms become similar, as expected. Student's t–test was used to verify the hypothesis that the use of negotiation speeds up the learning process. The value of T was computed for every game and the results showed that NQL is better than Q-learning when both are playing against a random opponent up to the 5000^{th} game, after which the results are comparable, with a level of confidence greater than 95%.

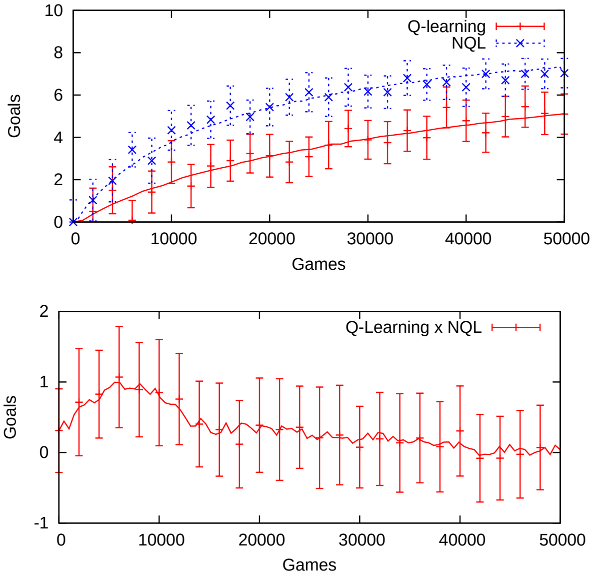

Figure 1: (top) Average goal difference for the Q-Learning and NQL learning against a random opponent and (bottom) for the Q-Learning versus NQL.

A second experiment tested the NQL when learning while playing against an opponent using Q-learning. Figure 1 (bottom) presents the learning curve (average of 30 training sessions, for $50,000$ games) for this experiment, where it can be clearly seen that the NQL algorithm is better at the beginning of the learning process and that after the a certain number of games the performance of this team becomes similar to the Q-learning, since all algorithms converge to equilibrium.

4. CONCLUSION

The experimental results obtained showed that the algorithm that use negotiation learned faster than in the case in which negotiation is not use. Future works include working on obtaining results in more complex domains, such as RoboCup Simulation and Small Size League robots, and applying this technique to other MARL algorithms.

5. REFERENCES

[1] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI '98*, pages 746–752,1998. AAAI.

[2] A. Fabregues and C. Sierra. An agent architecture for simultaneous bilateral negotiations. In *8th European Workshop on Multi-Agent Systems*, Paris, 2010.

[3] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ICML'94*, pages 157–163, 1994.