Learning against sequential opponents in repeated stochastic games

Pablo Hernandez-Leal
Centrum Wiskunde & Informatica
Science Park 123, Amsterdam, The Netherlands
Pablo.Hernandez@cwi.nl

Michael Kaisers
Centrum Wiskunde & Informatica
Science Park 123, Amsterdam, The Netherlands
Michael.Kaisers@cwi.nl

Abstract

This article considers multiagent algorithms that aim to find the best response in strategic interactions by learning about the game and their opponents from observations. In contrast to many state-of-the-art algorithms that assume repeated interaction with a fixed set of opponents (or even *self-play*), a learner in the real world is more likely to encounter the same strategic situation with changing counter-parties. First, we present a formal model of such *sequential interactions*, in which subsets from the player population are drawn sequentially to play a repeated stochastic game with an unknown (small) number of repetitions. In this setting the agents observe their joint actions but not the opponent identity. Second, we propose a learning algorithm to act in these sequential interactions. Our algorithm explicitly models the different opponents and their switching frequency to obtain an acting policy. It combines the multiagent algorithm PEPPER for repeated stochastic games with Bayesian inference to compute a belief over the hypothesized opponent behaviors, which is updated during interaction. This enables the agent to select the appropriate opponent model and to compute an adequate response. Our results show an efficient detection of the opponent based on its behavior, obtaining higher average rewards than a baseline (not modelling the opponents) in repeated stochastic games.

Keywords: Multiagent learning;
reinforcement learning;
repeated stochastic games

Acknowledgements

This research has received funding through the ERA-Net Smart Grids Plus project Grid-Friends, with support from the European Union’s Horizon 2020 research and innovation programme.

1 Introduction

Learning to act in multiagent systems has received attention mainly from game theory, focused on algorithms that converge to the Nash equilibrium [2, 6, 7]; and from reinforcement learning, focused on acting optimally in stochastic scenarios with limited a priori information but online observations [5, 11]. However, results are typically based on the assumption of self-play [3,7], i.e., all participants use the same algorithm, and a long period of repeated interactions.

In contrast, we focus on *sequential interactions* (as a variation of repeated stochastic games), i.e., the agent is paired with stochastically drawn opponents, with whom the agent interacts in short periods while observing joint actions, but without observing the opponent's identity. This article tackles the problem of finding the best response in this setting, which requires to hypothesize possible opponent types, and to identify opponents from their behavior. We propose an algorithm that supposes and identifies several common reasoning strategies, and which best-responds in face of residual uncertainty.

Related work in repeated stochastic games has proposed different learning algorithms [8, 9]. However, it is an open problem how to act quickly and optimally when facing different opponents, especially when uninformed about such changes. Bayesian and type based approaches are a natural fit for this kind of setting. A general and highly complex model are I-POMDPs [12] which are a framework for decision making in uncertain multiagent settings. Bayesian Policy Reuse (BPR) [15] is a framework for acting quickly in single-agent environments, assuming the availability of a policy library and prior knowledge of the performance of the library over different tasks. BPR computes a belief over *tasks* that is updated at every interaction and which is used to select the policy that maximises the expected reward given the current belief. Similarly, in this article the agent maintains believes over *opponent models*. Recently, work on Stochastic Bayesian Games [1] has compared several ways to incorporate observations into beliefs over opponent types when those types are re-drawn after every state transition. Our approach is similar in believe structure, but in contrast assumes that opponents are only redrawn after several repeated stochastic games.

Changes in the environment are also modeled explicitly by learning algorithms for non-stationary environments. Their goal is to learn an optimal policy and at the same time detect when the environment has changed, updating the acting policy accordingly. Reinforcement Learning with Context detection (RL-CD) [10] works in single agent tasks with a changing environment, i.e., different *contexts*. RL-CD learns a model of the specific task and assumes an environment that changes infrequently. To detect a new context RL-CD computes a quality measure of the learned models. Similar problems have been tackled in two-player repeated normal-form games, assuming an opponent with different stationary strategies to select from during a repeated interaction [13, 14]. The learning agent needs to learn online how to act optimally against each strategy while detecting when the opponent changes to a different one. While these works might be the closest state of the art, these approaches do not consider repeated stochastic games, which are central to our discussion.

We contribute to the state of the art with a framework that we name Opponent Learning in Sequential Interactions (OLSI), for best responding in repeated stochastic games by maintaining a hypothesized opponent model set. We compute a belief over this set, updating the belief and the set based on observations. The belief provides a way to compute future expected rewards more accurately and improves empirical performance, in our evaluation approaching optimality.

2 Problem setting

In contrast to classical RL, which considers one single agent in a stationary environment, Game theory studies rational decision making when several agents interact in a strategic conflict of interest, formalized as a *Game*. Note that different areas provide different terminology. Therefore, we will use the terms player and agent interchangeably and we will refer to other agents as opponents irrespective of the domain's or agent's cooperative or adversarial nature.

Our exposition of the approach is build on games with two players i and $-i$. A stochastic game comprises a set of stage games S (also known as states). In each stage s all players choose their part of the joint action $\mathbf{a} \in A(s)$. A game begins in a state $s_b \in S$. A joint action $\mathbf{a} = (a_i, a_{-i})$ is played at stage s and player i receives an immediate reward $r_i(s, \mathbf{a})$, the world transitions into a new stage s' according to the transition model $T(s, s', \mathbf{a})$. Reaching any goal state $s_g \in S$ terminates the game. The accumulated reward of a game is called an *episodic* reward.

We formalize *Sequential Interactions* (SI) as a specific variation of repeated stochastic games, where at each episode $k \in \{1, 2, \ldots, K\}$ a process draws a set of players $P_k \subset I$ from the population of individuals I to play a finite stochastic game that yields a reward (accumulated over the game) to each player. After the stochastic game terminates, the subsequent interaction commences. We specifically discuss the setting where the selection process is *stochastic* (as opposed to being a *strategic choice* by the agents), and the population comprises an unknown distribution over types of strategies. We consider P_k and opponent rewards within the stochastic game to be unobservable, while the joint actions are observable. In addition, neither $r_i(s, \mathbf{a})$ nor $T(s, s', \mathbf{a})$ are known. In our approach, the agent learns explicit models of $r_i(s, \mathbf{a}), T(s, s', \mathbf{a})$ and P_k from observations, and may update its hypothesized model of I.

3 Opponent learning in sequential interactions

Our approach is based on reinforcement learning and combines Bayesian beliefs over opponent types with PEPPER [8], which in turn builds on R_{MAX} [4]. We follow the assumptions of these state-of-the-art algorithms such as observing local immediate rewards but not the opponents', and knowing the maximum possible reward R_{max} for each episode.

The expected future rewards for a joint action **a** being in state s are defined by:

$$R(s, \mathbf{a}) = r(s, \mathbf{a}) + \sum_{s' \in S} T(s, s', \mathbf{a}) V(s') \tag{1}$$

where $V(s')$ is the expected future rewards of being in state s'. Note that given $r(\cdot), T(\cdot)$ and $V(\cdot)$, value iteration can be used to compute Equation 1. Note that these terms are defined in terms of the joint action, and therefore can be combined with various opponent models to compute expectations and best response policies.

R_{MAX} uses the principle of optimism in face of uncertainty, i.e., initially all states are assumed to result in maximal reward. To learn $r(\cdot)$ and $T(\cdot)$, Pepper uses frequencies of past games and we compute future rewards $V(s)$, by combining off-policy (e.g., Q-learning) and on-policy methods. This is, an on-policy estimation based on the observed distribution of joint actions, using $n(s), n(s, \mathbf{a})$ for the number of visits to state s and the number of times joint action **a** was chosen in that state respectively. Then, the on-policy estimation is given by: $V^{on}(s) = \sum_{\mathbf{a} \in A(s)} \frac{n(s,\mathbf{a})}{n(s)} R(s, \mathbf{a})$ and the combined estimation $V(s) = \lambda(s)\hat{V}(s) + (1 - \lambda(s))V^{on}(s)$. Where $\hat{V}(s)$ represents an optimistic approximation given by $\hat{V}(s) = \max(V^{\text{off}}(s), V^{\text{on}}(s))$ and where $\lambda \in [0, 1]$ represents a stationarity measure initialized to one but approaching zero when the agent gets more experience.[1] We use the concept of non-pseudo stationary restarts [8], i.e., when $R(s)$ is observed to not be pseudo stationary $\lambda(s)$ resets to one. Let $n'(s)$ be the number of visits to stage s since $R(s)$ was last observed to not be pseudo-stationary, then: $\lambda(s) = \max\left(0, \frac{C - n'(s)}{C}\right)$ with $C \in \mathbb{N}^+$.

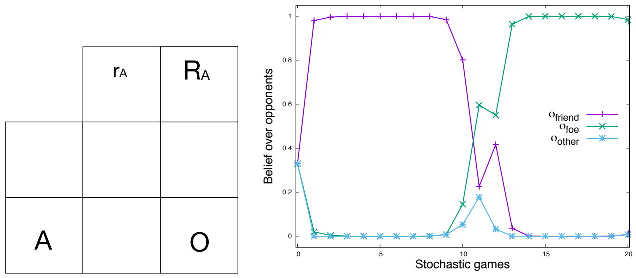
Algorithm 1: Opponent learning in sequential interactions (OLSI) framework

```
Input: Maximum possible reward $R_{max}$, opponent space $\mathcal{H}$, prior probabilities $P(\mathcal{H})$, switch frequency probability $P(s|n)$
1  Initialize beliefs $\beta^0(\mathcal{H}) = P(\mathcal{H})$
2  Initialize $V(\cdot)$ with $R_{max}$
3  Random initial policy $\pi$
4  for each stochastic game $k$ do
5      Update $R(\cdot)$; see Eq. 1
6      Update policy $\pi$ according to $\beta^k(h)$
7      Observe state
8      while state is not goal do
9          Select action $a$
10         Observe state and opponent action $a_{-i}$
11         Receive immediate reward $r$
12         Compute opponent model $\pi_h$ for each opponent $h \in \mathcal{H}$
13         Update belief with observation $a_{-i}$ as $\beta^k(h) = \frac{P(a_{-i}|h)\beta^{k-1}(h)}{\sum_{h' \in H} P(a_{-i}|h')\beta^{k-1}(h')}$
14         if enough visits to $(s, \mathbf{a})$ then
15             Update rewards, $V(\cdot)$ and transition model
16             Update $R(\cdot)$; see Eq. 1
17             Update policy $\pi$ according to $\beta^k(h)$
18     $\beta^{k+1}(h) = P(s|n)\beta^k(h) + (1 - P(s|n))P(\mathcal{H})$
```

Our proposed framework OLSI is described in Algorithm 1. Note that removing the lines that correspond to the belief update (lines 1, 13 and 18) and assuming only one opponent model to compute the policy (lines 6, 12, and 17) yields Pepper. Our framework thus generalizes Pepper by taking into account multiple different hypothesized opponent models (policies) at the same time, computing the agent's best response against the belief over them.

Consider the following common opponent policies: 1) a totally cooperative (friend) opponent, who play its part of maximizing the agent's reward (e.g., due to identical rewards for both): $\pi_{friend} = \max_{\mathbf{a}_{-i}} R(s, \mathbf{a})$, or 2) a competitive (foe) opponent that aims to minimize the agent's reward (e.g., in zero-sum games): $\pi_{foe} = \min_{\mathbf{a}_{-i}} R(s, \mathbf{a})$. Additionally, the agent can 3) learn by observations $\pi_{freq} = \frac{n(s, a_{-i})}{n(s)}$, accounting for mixed types. Note that if the opponent model is correct, it can be used to compute $R(s)$ that accurately reflects future rewards that only depend on the agent's action.

[1] Recall that $R(s, a)$ is initialized to R_{max} so it is likely to decrease in early episodes, but eventually will become pseudo-stationary.



(a) (b)

Figure 1: (a) A stochastic game with two players, the learning agent (A) and one opponent (O). The learning agent receives a reward when it reaches state marked with r_A or R_A with $r_A < R_A$. In case of collision the opponent has priority over the state. (b) OLSI's belief over the three opponent types, from games 1-10 OLSI is facing O_{friend}, and then O_{foe} for the remaining games. OLSI's belief accurately represent the opponent identity by being updated at every game and also taking into account the switching frequency (10 in this case).

The belief is updated within a single stochastic game using the observation a_{-i} (opponent action) and the probability of that action being generated by each opponent policy using Bayes' theorem: $\beta^k(h) = \frac{P(a_{-i}|h)\beta^{k-1}(h)}{\sum_{h' \in H} P(a_{-i}|h')\beta^{k-1}(h')}$.

Additionally, OLSI also takes into account the switching frequency of the opponents to update the belief when a game ends. For this, the agent maintains a probability for the event opponent switch, $P(s|n)$, given that n games have been played since the last switch. Then, we update the belief as follows: $P(s|n)\beta^{k-1}(h) + (1 - P(s|n))P(\mathcal{H})$. This is, we consider the probability of facing the same opponent (keeping the same belief) or if a switch is likely to happen then the belief moves to a more conservative prior distribution.

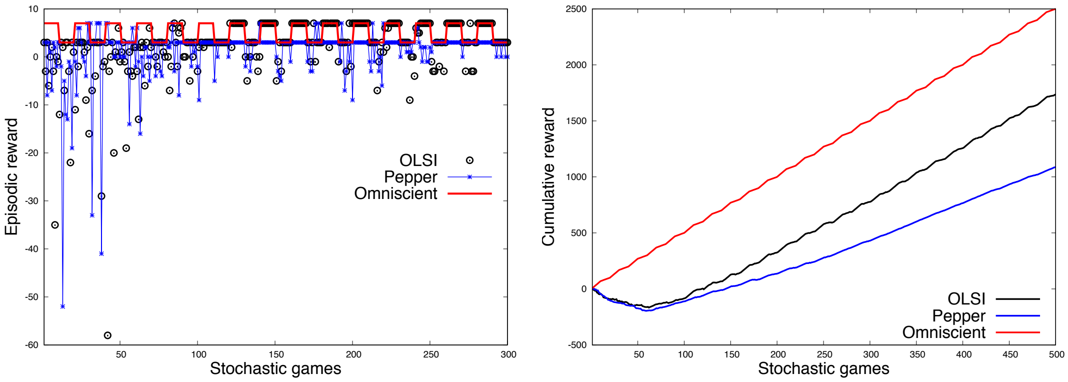
4 Experiments

For the sake of clarity we selected a simple setting with intuitive opponents that simplify the interpretation of the behavior and results. We evaluated our proposed algorithm on a stochastic game represented as a grid world. We performed experiments comparing OLSI, Pepper (with only one opponent model based on frequencies of opponent actions) and an Omniscient agent that knows how to best respond at all times. Figure 1 (a) depicts a graphical representation of the stochastic game used in the experiments. There are two players, the learning agent (A) and the opponent (O). The starting positions are marked with their initial. The learning agent receives a reward when it reaches a goal state r_A or R_A, with $r_a < R_A$. The agents can move upwards or horizontally, and the opponent has the possibility to stay in the same state; the learning agent moves always to the right and the opponent to the left; to avoid agents getting trapped the grid is a toroidal world. With every step that does not transition to a goal state the learning agent receives a penalty p_{min}. In case of collision the learning agent receives high penalty p_{max} with $p_{min} < p_{max}$.

Note that the opponent directly influences the possible reward the learning agent can obtain. For example, since the opponent is closer to R_A it can block its way to the learning agent, in which case the best option would be to go for r_A. For the experiments we set $r_A = 5, R_A = 10, p_{min} = -1, p_{max} = -5$.

We test against a population with two types of opponents, a *foe* opponent, O_{foe}, that aims to block the way to R_A and a *friendly* opponent, O_{friend}, that allows you to pass to R_A. First, we present how the belief behaves in a sequential interaction of 20 stochastic games; during the first 10 games the agent faces O_{friend}, thereafter the agent is paired with O_{foe}. Figure 1 (b) depicts the belief of the OLSI agent, which assumes a opponent space composed of three opponents $\{O_{friend}, O_{foe}, O_{other}\}$, additionally we modelled the switch frequency using a Gaussian distribution $\mathcal{N} = (10, 1)$. OLSI starts with a uniform prior over the three possible opponents and when the first game ends the belief for O_{friend} increases and remains until round 9 where it slightly decreases since the switch frequency is getting closer to the expected value of n. The opponent changes to O_{foe} at game 11, which is reflected in OSLI's belief.

Figure 2 (a) compares episodic rewards of OLSI against Pepper and an Omniscient agent in a sequential interaction of 200 stochastic games. The opponent alternates between O_{friend} and O_{foe} every 10 games. From the figure it can be noticed that approximately in the first 100 games both OLSI and Pepper are learning and obtaining low rewards in some games. However, after this period OLSI learns a best response against each of the two possible opponents obtaining R_A against the friendly opponent and r_A against the foe opponent. In contrast, Pepper learns a safer policy in which in most of the times it opts for the r_A reward even when facing the friendly opponent. Figure 2 (b) depicts cumulative rewards of the compared approaches over 10 repetitions of 500 stochastic games against the same alternating opponents. Results



(a) (b)

Figure 2: (a) Episodic rewards of OLSI, Pepper and the Omniscient agent against the same alternating opponents. Pepper learns to collect r_A as a safe response against the population; in contrast, OLSI best responds by identifying the switching opponents. (b) Corresponding cumulative rewards show OLSI's asymptotic performance parallels the Omniscient's.

show that after the learning phase Pepper obtains lower rewards than the Omniscient agent while OLSI achieves parallel asymptotic performance.

5 Conclusions

This article complements the body of literature on learning algorithms for multiagent systems, which often assumes self-play or stationary opponents, by considering the scenario of repeated stochastic games against switching opponent sets. Our first contribution is the formalisation of this setting of sequential interactions. Our second contribution is a framework, OLSI, for learning in that setting based on reinforcement learning and Bayesian beliefs. OLSI maintains a belief over the hypothesized opponents, which is used to compute an adequate response. Results in a simple scenario showed that OLSI achieved near-optimal results, thus outperforming the conservative strategy learned with a single opponent model. These promising preliminary results suggest to evaluate the effectiveness of OLSI and the effect of parameters (e.g., priors) on algorithm performance in a wider set of experiments with more complex opponents. Once results are validated against further benchmarks, this approach may boost reinforcement learning performance in a broader set of realistic strategic interaction scenarios against changing counter-parties.

References

[1] S. V. Albrecht, J. W. Crandall, and S. Ramamoorthy. Belief and truth in hypothesised behaviours. *Artificial Intelligence*, 235:63–94, 2016.

[2] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers. Evolutionary Dynamics of Multi-Agent Learning: A Survey. *Journal of Artificial Intelligence Research*, 53:659–697, 2015.

[3] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

[4] R. I. Brafman and M. Tennenholtz. R-MAX a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

[5] L. Busoniu, R. Babuska, and B. De Schutter. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[6] D. Chakraborty and P. Stone. Multiagent learning in the presence of memory-bounded agents. *Autonomous Agents and Multi-Agent Systems*, 28(2):182–213, 2013.

[7] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2006.

[8] J. W. Crandall. Just add Pepper: extending learning algorithms for repeated matrix games to repeated markov games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 399–406, Valencia, Spain, 2012.

[9] J. W. Crandall. Robust Learning for Repeated Stochastic Games via Meta-Gaming. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 3416–3422, Buenos Aires, Argentina, 2015.

[10] B. C. Da Silva, E. W. Basso, A. L. Bazzan, and P. M. Engel. Dealing with non-stationary environments using context detection. In *Proceedings of the 23rd International Conference on Machine Learnig*, pages 217–224, Pittsburgh, Pennsylvania, 2006.

[11] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian Reinforcement Learning: A Survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

[12] P. J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*, 24(1):49–79, 2005.

[13] P. Hernandez-Leal, M. E. Taylor, B. Rosman, L. E. Sucar, and E. Munoz de Cote. Identifying and Tracking Switching, Non-stationary Opponents: a Bayesian Approach. In *Multiagent Interaction without Prior Coordination Workshop at AAAI*, Phoenix, AZ, USA, 2016.

[14] P. Hernandez-Leal, Y. Zhan, M. E. Taylor, L. E. Sucar, and E. Munoz de Cote. Efficiently detecting switches against non-stationary opponents. *Autonomous Agents and Multi-Agent Systems*, 2016.

[15] B. Rosman, M. Hawasly, and S. Ramamoorthy. Bayesian Policy Reuse. *Machine Learning*, 104(1):99–127, 2016.