Skill Reuse in Partially Observable Multiagent Environments

Double Blind Review

1 Introduction

Learning to act in multiagent systems is distinctive from traditional single-agent learning, in that the optimal strategy depends on others: an agent seeks to find the best response strategy and the other agents may adapt their strategies in turn. This problem has been approached from several angles with different assumptions from game theory to deep reinforcement learning (RL) [1, 4, 8, 7, 12]. However, many algorithms require a long period of repeated interaction to learn appropriate responses, which may be unrealistic in many domains. Also, agents can change their strategies, rendering the learned model irrelevant. One last difficulty is that many domains are only *partially observable* [9], meaning that the agent cannot observe the environment (and the opponents) all the time. Therefore, our goal is to tackle partially observable multiagent scenarios by proposing a framework based on learning robust best responses (i.e., skills [11]) and Bayesian inference for opponent detection [14]. In order to reduce long training periods, we propose to intelligently reuse policies (skills) by quickly identifying the opponent we are playing with. To quickly adapt to non-stationary learning opponents, we assume that their strategies can be clustered, which we can then best respond against [3, 6, 13]. Then, our solution is a two step process of (1) generating robust skills (offline), and (2) reusing those in an online manner.

2 Robust and reusable skills

Learning skills in multiagent environments Q-learning is a well-known algorithm for RL [17], however, in practice it can take significant amounts of data to converge. In contrast, *n-step* methods or eligibility traces usually show better performance [15], e.g., True Online Sarsa(λ) [16] was used as base algorithm for learning the policies (best responses to opponent strategies). Since we aim for *robust* policies across environments we take insights from single-agent learning by incorporating the *skills* concept. Skills' core idea (in single-agent RL) is that they retain the same semantics across tasks [11]. Therefore, we use features that are *relative* to the learning agent, which makes them robust to changes in the environment.

We formalize the concept of skill in multiagent environments as follows, for agent i with respect to opponent $-i$, a skill is a 3-tuple $\langle \mathcal{I}^o_{i,-i}, \pi^o_{i,-i}, \mathcal{T}^o_{i,-i} \rangle$, where $\mathcal{I}^o_{i,-i} : (s|\pi_{-i}) \rightarrow \{0,1\}$ is the initiation set (where the skill can be used), the skill policy $\pi^o_{i,-i} : (s, a_i|\pi_{-i}) \rightarrow [0,1]$ defines how to act, and a termination condition, $\mathcal{T}^o_{i,-i} : (s|\pi_{-i}) \rightarrow \{0,1\}$, determines when to stop using the skill. Note that skill policies are conditioned on the opponent since they represent a best response.

What is missing is to determine what opponent strategies should be considered. We define a *hypothesized opponent policy (HOP)* set; for example, in poker, experts describe strategies based on few features [13]. In general, agents need to interact with opponents that have not only one but a set of different behaviors. For best performance, the agent needs to know how to respond with appropriate policies for each different opponent behavior while also continuously estimating possible changing opponent behaviour. In our experiments we assume the opponent can use two possible strategies, being offensive or defensive.

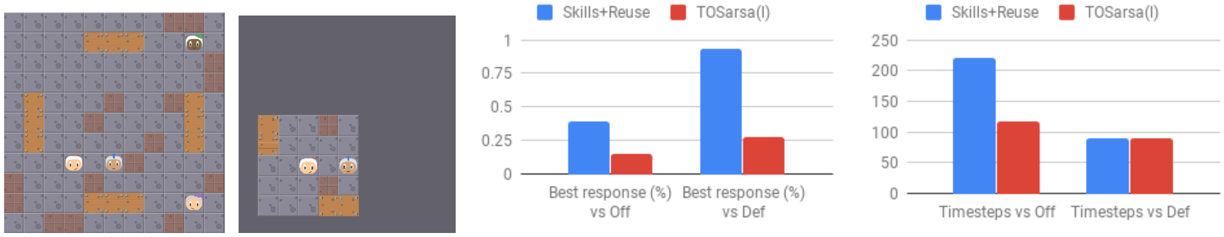

Figure 1: (Left) Pommerman board example, grey cells are passages, light brown cells are wood and dark brown cells are rigid walls; (Center) partial views of an agent. (Right) Our learning agent is capable of identifying the opponent and best responding in partially observable randomly generated boards, in contrast to a trained single-agent RL algorithm which is best responding to the mixed behavior of the opponent. When facing against a defensive opponent, our agent takes ≈ 100 timesteps to kill the opponent, with higher efficiency ($\approx 90\%$) than the RL agent ($\approx 25\%$), and when facing an offensive opponent stays alive twice the time, relative to the RL agent (results of 10,000 games).

Reusing skills with opponent tracking Identifying, tracking and predicting changing behaviors is an open challenge for autonomous agents [4, 5]. If the HOP set is known, together with their respective best responses, then identifying the current opponent policy from the HOP set is all it takes to be optimal. Moreover, if the HOP is known, the agent can learn *observation* models ahead of interaction. These performance models can then be used online to infer a belief over the HOP.

The high level idea of the online tracking algorithm is the following: (1) when the agent is in a *strategic interaction (SI)*, i.e., there is an opponent in view, the agent gets an observation σ (e.g., opponent action), (2) which is used to update the belief, β, over the type of SI the agent is facing. For a set of previously solved SI, $\mathcal{H}$, and a new instance $h^\star$, the *belief* β is a probability distribution over $\mathcal{H}$ that measures to what extent $h^\star$ matches the known strategic interactions in their observation signals [10, 14]. (3) The belief is used to obtain the most probable SI, $arg \max \beta$, we assume the behavior of the opponent(s) is prescribed by π_{-i} for the current SI, and therefore the best response is readily available $\pi = BR(\pi_{-i})$.

3 Experimental results

Setup The Pommerman environment [2] is played on an grid with up to four agents, each with six possible actions: four movement actions, do nothing, or place a bomb. Each cell on the grid is a passage, a rigid wall or wood. When an agent places a bomb, it will explode after 10 timesteps and its blast destroys wood and any agents in its way. Maps are randomly generated on every episode. In the partially observable version of the game, only a surrounding area of an agent is visible (see Figure 1). In our experiments we used two agents, one learning agent and one opponent. The opponent has two fixed strategies to follow: an offensive one (rule-based) that places bombs, and a defensive one that does not place bombs. Note that the best response of the learning agent depends on the opponent strategy: staying alive for at least 300 timesteps against the offensive strategy, and attacking (killing the opponent) against the defensive one. The board is set to 8×8 and is randomly shuffled for every game in our experiments. The opponent randomly chooses a strategy on every game. Then, our learning agent needs to identify the opponent strategy and best respond in a partially observable environment.

Results We compared our approach, Skills+Reuse, to a trained True Online Sarsa(λ) agent. Our preliminary results suggests that: (1) it is possible to generalize skills to multiagent environments; (2) while dealing with unknown opponents Bayesian inference can be used in partially observable scenarios; and (3) opponent behaviour identification can yield better results than a single-agent RL since it can best respond to specific opponents, instead to a mixed behavior (see Figure 1, right).

References

[1] Open AI Five. `https://blog.openai.com/openai-five`, 2018. [Online; accessed 7-September-2018].

[2] Pommerman. `https://www.pommerman.com/`, 2018. [Online; accessed 7-September-2018].

[3] Stefano V. Albrecht, Jacob W. Crandall, and Subramanian Ramamoorthy. Belief and truth in hypothesised behaviours. *Artificial Intelligence*, 235:63–94, 2016.

[4] Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, February 2018.

[5] Tim Baarslag, Michael Kaisers, Enrico H Gerding, Catholijn M Jonker, and Jonathan Gratch. Computers That Negotiate on Our Behalf - Major Challenges for Self-sufficient, Self-directed, and Interdependent Negotiating Agents. *AAMAS Workshops*, 10643 LNAI(2):143–163, 2017.

[6] Nolan Bard, Deon Nicholas, Csaba Szepesvári, and Michael Bowling. Decision-theoretic Clustering of Strategies. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 17–25, Istanbul, Turkey, May 2015.

[7] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. Evolutionary Dynamics of Multi-Agent Learning: A Survey. *Journal of Artificial Intelligence Research*, 53:659–697, 2015.

[8] Lucian Busoniu, Robert Babuska, and Bart 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.

[9] Anthony R. Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes*. PhD thesis, Computer Science Department, Brown University, May 1998.

[10] Pablo Hernandez-Leal, Matthew E. Taylor, Benjamin Rosman, L. Enrique Sucar, and Enrique 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.

[11] George Konidaris and Andrew G. Barto. Building Portable Options - Skill Transfer in Reinforcement Learning. *International Joint Conference on Artificial Intelligence*, pages 895–900, 2007.

[12] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163, New Brunswick, NJ, USA, 1994.

[13] Marc Ponsen, Karl Tuyls, Michael Kaisers, and Jan Ramon. An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing*, 1(1):39–45, January 2009.

[14] Benjamin Rosman, Majd Hawasly, and Subramanian Ramamoorthy. Bayesian Policy Reuse. *Machine Learning*, 104(1):99–127, 2016.

[15] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[16] Harm van Seijen, Ashique Rupam Mahmood, Patrick M Pilarski, Marlos C Machado, and Richard S. Sutton. True Online Temporal-Difference Learning. *Journal of Machine Learning Research*, 2016.

[17] John Watkins. *Learning from delayed rewards*. PhD thesis, King's College, Cambridge, UK, April 1989.