Learning in the Multi-agent Setting

Anonymous ID: 6
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4

December 22, 2011

Abstract

The study of multi-agent systems investigates the interaction between agents in the setting where multiple agents exist and their behaviour affects each other's. Because in most environments agents are not likely to know ahead of time what other agents will do, what the world will be after the agents' joint behaviour, or sometimes even the existence of other agents, it is reasonable that they will try to *learn* from history so as to act in a way that is most beneficial to them. Multi-agent learning (MAL) has emerged as a field at the intersection of multi-agent systems and machine learning. It concerns the application of learning techniques to the multi-agent setting. Over recent years, MAL has drawn significant attention from the research communities of game theory and artificial intelligence (AI). This paper presents a survey of recent work in the literature and integrates some comment and remarks from different papers in an organized structure. Beyond this, we also discuss and compare the different remarks, and offer our own thoughts about the work we survey and where the literature might head in the future.

1 Introduction

Machine learning has seen a huge success in various domains, both in the academic community where computational models have been built to advance the artificial intelligence research and in many commercial services or products where ranking algorithms and recommender systems are capable of providing helpful information to the customers. It is thus a natural move to introduce *learning agents* in the multi-agent setting. *Multi-agent learning* (MAL) has emerged as a field at the intersection of multi-agent systems and machine learning. From a simple viewpoint, it just concerns the application of learning techniques to the multi-agent settings. Nevertheless, even the attempts to apply learning techniques seem to pose a considerable number of problems, among which one of the most fundamental, as argued by Shoham et al., is the evaluation criteria [7]. That is, after applying some learning rule in a multi-agent system, how do we assess and justify its success or simply make sense of it? In this paper, we present a survey of recent work in the literature of MAL, including the introduction of some most common and widely studied MAL techniques. We note here that the survey presented in the paper is far from comprehensive of the whole literature and is largely based on the work in [1, 5, 6, 7], most of which are also surveys or introductions to the field of MAL. However, we emphasize more on the remarks that other researchers have about recent work in MAL and their relations to other fields of study. In particular, Shoham et al. identify distinct goals pursued in MAL research [7], and Frudenberg et al. comment on their work from an economist's perspective [4]. We involve ourselves in the discussion and offer our thoughts when comparing their comment and observations. In addition, whenever appropriate we try to draw relevance to general machine learning techniques or concepts.

In presenting the survey, we take a rather game-theoretic perspective, just as what most work in the literature does. In the next section, we give the necessary background in *learning* and *game-theoretic formalism*. In Section 3, we survey some most widely studied learning techniques in multi-agent systems. Instead of having a separate section for all the comment and discussion, we discuss different researchers' comment and offer our own where appropriate along the way. Similar to most other surveys, our presentation remains a relatively high-level view of the work rather than delving into the detailed technicality.

2 Background

In this section, we provide some necessary vocabulary and definitions in *game theory*, based on which the work in the literature is presented. We also give a brief overview of *learning*, including the various possible issues in multi-agent settings that make MAL intrinsically more complicated.

2.1 Game-theoretic formalism

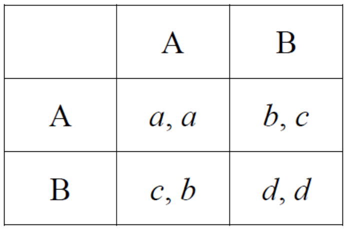
Here we formalize the setting from a game-theoretic perspective, in which we present and discuss the multi-agent learning techniques. We ground our discussion in the simplest *normal-form game*, in which all agents simultaneously take an action and end up in a certain outcome according to their joint actions. Then each agent receives a payoff based on his utility function and the outcome. Despite its simplicity, many scenarios in the multi-agent setting can be described by a series of the one-shot normal-form games, and it is also the framework in which most work in the multi-agent systems take place. We now define a *stochastic game*, which is a series of normal-form games.

Definition A *stochastic game* is a tuple (N, S, A, U, P), where N denotes a set of n agents, S a set of n-agent normal-form stage games, $A = (A_1, A_2, ..., A_n)$ with A_i specifying the set of actions available for agent i (Here we implicitly assume that each agent has the same action space in all stage games), $U = (U_1, U_2, ..., U_n)$ with $U_i : S \times A \rightarrow \Re$ giving agent i's immediate payoff function for each stage game, and $P : S \times A \rightarrow \Pi(S)$ specifying the transition probabilities after each stage game.

When there is only one stage game in S, a stochastic game reduces to a *repeated game* (and in this case, the transition probabilities are irrelevant). On the other hand when there is only one agent at play, a stochastic game reduces to a *Markov decision problem* (MDP), which has been the fundamental setting leading most of the work in artificial intelligence.

We then define a *symmetric game* and introduce the term *self-play* as we shall refer to both in the following discussion of learning in games.

Definition A *symmetric game* is two-player normal-form game of the form shown in Figure 1, where both players have the same strategy space (but they can in fact have more than two actions), and the payoff for playing a particular strategy is independent of which player does so.



	A	B
A	a, a	b, c
B	c, b	d, d

Figure 1: Two-player symmetric game: the payoff for playing a particular strategy is independent of the player.

Terminology *Self-play* refers to the situation in which all agents employs the same learning mechanism in a multi-agent system.

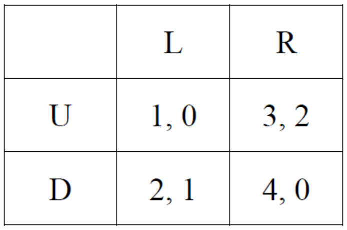
We highlight and hope it is obvious that self-play certainly does not mean that all agents always perform the same action in each stage game.

2.2 Overview of learning

While many single-agent learning algorithms have proved successful in the field of artificial intelligence, their usefulness in the multi-agent setting can be hardly justified. In some context, it is even quite difficult to define what constitutes learning. From the viewpoint of machine learning (in the single-agent setting), most often learning concerns the ability to predict. We normally identify ”learning” if one is capable of, based on what he has observed in the history (e.g. training examples), correctly predicting the future (e.g. unseen test examples). This actually shares the same concept with Foster et al.'s definition in the game-theoretic setting that players ”learn” if they eventually succeed in predicting their opponents' behaviour with high degree of accuracy [2]. However, the concept of learning becomes much more complicated in the multi-agent setting. The fundamental underlying reason is that in the existence of other agents who are also learning, an agent's learning process will affect the other agents' and vice versa. To paraphrase Young's words, when agent A is trying to learn about agent B and behave according to what A learns. Since B is also learning and observes A's behaviour, B's behaviour can change as a result of A's attempts to learn it [10].

Another source of the conceptual complexity of MAL is the inseparability of *learning* and *teaching* [7]. That is, in the multi-agent setting, agents are inherently learning and teaching at the same time. To make the concept more concrete, consider the repeated version of the normal-form stage game shown in Figure 2.

In the stage game, the row player has a strictly dominant strategy D (and so U is strictly dominated). If the row player always plays his dominant strategy in the repeated game, the column player will most likely continue responding with L, ending up in the outcome (D, L) forever. However, if the row player repeatedly plays U for a long while, the column player will eventually start responding with R, resulting in the outcome (U, R), which is better for both players. In this case, knowing that the column player's behaviour depends on his learning about the row player, the row player can actually *teach* the column player by playing his dominated strategy in the stage game. On the other hand, Fudenberg et al. also argue that this kind of teaching plays no role in



	L	R
U	1, 0	3, 2
D	2, 1	4, 0

Figure 2: Stackelberg stage game: the first number in each cell denotes the payoff of the row player, whereas the second number represents that of the column player.

certain environments [4]. Consider, for example, that in a large but finite population, each agent is repeated matched with another at random to play a single-shot normal-form game, and the only thing they observe is the actions and payoffs in the match. If the population is sufficiently large, then the agent is unlikely to be matched with the current opponent for a long time. Therefore, the ”teaching” does not appear very relevant in the case. The key idea we highlight here is that the issues in MAL depend heavily on the context in which the problems are addressed, and we must be very cautious when considering them.

The context also tells how we should model different scenarios in the multi-agent setting, including whether agents can observe their own payoffs, whether they can observe the opponents' payoffs or actions, and whether they know the transition probabilities in the case of a stochastic game. In the rest of the paper, we concentrate on the setting in which the payoffs and actions are observable by all the agents since much of the current work on MAL adopts this setting, and many scenarios can also be reasonably modelled in this setting. We also restrict the discussion in two-player games for simplicity.

3 MAL in games

We follow Shoham et al.'s categorization of learning techniques depending on whether agents explicitly model the opponents' strategies when deciding what actions to take [5]. We describe some learning techniques in the two distinct categories: *model-based learning* and *model-free learning.* Then we introduce *social learning*, which concerns only population statistics under different learning rules, rather than considering all the individual behaviour and payoffs.

3.1 Model-based learning

In model-based learning, an agent assumes that the opponent uses a unknown stationary strategy which is fixed over repeated plays of the stage game. So the agent simply tries to model the opponent's fixed strategy based on the history of the repeated plays. The model-based learning has, essentially in Shoham et al.'s language [7], the following scheme:

1. Initialize some model of the opponent's strategy.

2. Play a best response to the assessed model of the opponent.

3. Observe the opponent's actual action and update the assessed model of his strategy.

4. Go to step 2.

One of the earliest and simplest model-based learning rules is the *fictitious play*, in which the model of the opponent's strategy purely corresponds to the frequency of each action played in the history. That is, the opponent's strategy is modelled as $(P(a_1), P(a_2), \ldots)$, where

$$P(a) = \frac{w(a)}{\sum_{a' \in A} w(a')} \tag{1}$$

and $w(a)$ denotes the number of times that action a has been played in the history. For instance, if the column player has played (R, R, L, R) in the first four rounds of the repeated game described in Figure 2, then in the fifth round, the row player will model the column player's strategy as the mixed strategy $(R : 0.75, L : 0.25)$. Another more complex form of model-based approach is *Bayesian learning.* Unlike fictitious play, in Bayesian learning scheme, agents do not model the opponents as having a stationary strategy but any repeated-game strategies. That is, agents' beliefs about the opponents can include any probability distribution over the set of possible repeated-game strategies. After each round of the stage game, agents use *Bayesian updating* to update their beliefs about the opponents: given the history h, agent i assigns a probability of the opponent playing a particular strategy $s_{-i} \in S_{-i}$ to be

$$P_i(s_{-i} \mid h) = \frac{P_i(h \mid s_{-i}) P_i(s_{-i})}{\sum_{s'_{-i} \in S_{-i}} P_i(h \mid s'_{-i}) P_i(s'_{-i})}. \tag{2}$$

We ourselves have some favour to Bayesian learning from the neuroscience's perspective since it has been studied and well accepted that Bayesian conditioning (on prior belief) indeed captures the way humans reason about the world.

Although Shoham et al. argue that those successful machine learning techniques in single-agent settings in AI should not be expected a priori to prove relevant in the multi-agent setting due to much of the inherent complicatedness of MAL [7], we believe these machine learning algorithms definitely will not be completely irrelevant. As such we suggest that more sophisticated statistical learning approach may find its stand in the multi-agent setting too. Here we adapt, without further analyzing, a statistical learning rule which tries to exploit the temporal locality in the history of the opponent's plays (this comes to our mind when surveying the literature) to the multi-agent setting. First, we specify a number k that measures the length of history the agents remember back from now. When modelling the opponent's strategy in the t^{th} round of the repeated game, an agent looks into the history, finds all instances of k consecutive plays matching that from round $t - k$ to round $t - 1$, and forms the frequency of actions played in the history conditioned on that the previous k plays match the pattern from round $t - k$ to round $t - 1$ as the opponent's strategy at round t. To illustrate with an example. Consider again the repeated version of the game shown in Figure 2. Assume the column player has an action history $h = (\ldots, R, L, R)$ and $k = 3$. To model the opponent's strategy, the row player finds all instances of 3 consecutive plays in h matching the pattern (R, L, R) and looks at the next play following the pattern in these instances, using these to form the conditional probability distribution as the opponent's strategy.

3.2 Model-free learning

Using model-free approaches, agents do not attempt to learn a model of the opponents' strategies but rather try to learn to maximize their own utilities over all possible actions over time. Before introducing any learning algorithm, we first cover the *value iteration* method for solving a known MDP (i.e. the payoffs and transition probabilities are available to the agent), based on which most learning techniques develop. The value iteration proceeds by iteratively updating a value function $V : S \to \Re$ with the Bellman equation:

$$V_{t+1}(s) \leftarrow \max_{a \in A} \{U(s,a) + \beta \sum_{\hat{s} \in S} P(s,a,\hat{s}) V_t(\hat{s})\}, \tag{3}$$

where β denotes the discount factor for future payoffs in the MDP. This method guarantees convergence to the optimal V^*, which gives the maximum utility value starting in the given state s (i.e. the starting stage game). Building on the basic idea of value iteration, the *Q-learning* algorithm can be used to solve for the optimal policy in a unknown MDP (i.e. payoffs and transition probabilities are unknown to the agent). With arbitrarily initialized functions $Q : S \times A \to \Re$ and $V : S \to \Re$, the algorithm proceeds by the iterative updates:

$$\begin{aligned} Q_{t+1}(s_t, a_t) &\leftarrow (1 - \alpha_t) Q_t(s_t, a_t) + \alpha_t [U(s_t, a_t) + \beta V_t(s_{t+1})] \\ V_{t+1}(s) &\leftarrow \max_{a \in A} Q_t(s, a) \end{aligned} \tag{4}$$

s_t and a_t are the current state at time t and the selected action at time t, respectively. α_t denotes the learning rate. Watkins et al. prove the convergence of the Q-learning algorithm to the Q^* and V^* values of the optimal policy if every action-state pair is eventually sampled infinitely many times, and the learning rate α_t satisfies certain reasonable constraints [8]. A lot of attempts have been made to extend the Q-learning to multi-agent stochastic games (recall that an MDP is a single-agent stochastic game), yet not too much success have been seen in general games. Nevertheless, researchers are able to use variants of Q-learning algorithms to solve some special cases of stochastic games, namely pure-coordination (a.k.a common-payoff) games and pure-competition (a.k.a zero-sum) games.

3.3 Social learning

We feel it is worth devoting a separate sub-section to the discussion of *social learning*, in which models are developed for the learning process of population of agents rather than that between individual agents. The term *social learning* is borrowed from the biological literature, and much related work in the MAL field is motivated by biological inspirations. In particular, we present the widely adopted *replicator dynamics*, a model originated from population biology to simulate the process of biological evolution. The model assumes a homogeneous population of agents, in which agents are continually paired at random to play a symmetric game, receiving some immediate payoffs. Then agents "reproduce" in proportion to the payoffs they receive in the game. The mathematical model is formally described as follows. Let S be the set of possible (pure) strategies and $u : S \times S \to \Re$ be the payoff function of a given symmetric game. The fraction of agents playing strategy s at time t is

$$\theta_t(s) = \frac{\phi_t(s)}{\sum_{\hat{s}\in S} \phi_t(\hat{s})}, \tag{5}$$

where $\phi_t(s)$ denotes the number of agents playing strategy s at time t. The expected payoff to any agent playing strategy s at time t is then

$$u_t(s) = \sum_{\hat{s}\in S} \theta_t(\hat{s})u(s, \hat{s}). \tag{6}$$

The change in population, or the reproduction rate, of agents playing strategy s at time t is defined to be proportional to the expected payoff of playing strategy s at current time, that is,

$$dt(\phi_t(s)) = \phi_t(s)u_t(s), \tag{7}$$

or equivalently,

$$\phi_{t+1}(s) = \phi_t(s)[1 + u_t(s)]. \tag{8}$$

From this formulation, we see that the number of agents playing some particular strategy will keep increasing as long as the expected payoff for playing that strategy is greater than zero. However, in this population model or any social learning scheme, we only concern the relative proportion of agents playing some particular strategy in the entire population. Differentiation on both sides of (5) results in the change in the fraction of agents playing strategy s at time t:

$$dt(\theta_t(s)) = \theta_t(s)[u_t(s) - \bar{u}_t], \tag{9}$$

where $\bar{u}_t = \sum_{s\in S} \theta_t(s)u_t(s)$ is defined to be the average expected payoff of the entire population at time t. We now notice that the fraction of agents playing a particular strategy in the population will only increase if the expected payoff for playing that strategy is greater than the current average expected payoff.

Although much of recent work in the MAL field does not focus on this framework, we believe that there are lessons to be learned (sorry for the pun) in the sense that the social learning scheme has successfully adopted biological inspiration. Just as that adaptation of biological models has contributed a great deal to the advances of machine leaning and AI in general (e.g. the well-known *neural network*) and indeed many researchers consider that further success of AI requires strong collaboration of machine learning, neuroscience, and other biological sciences, it seems quite possible that there is an important place for them in the multi-agent setting too. Alonso et al. mention some social learning mechanisms from biology that can be implemented in software agents, and we refer interested readers to [1].

4 Evaluation criteria

We have intentionally left out any evaluation of learning algorithms in the above presentation. Here we give a more concentrated discussion. Shoham et al. summarize some typical results in the literature regarding the MAL algorithms [7], and we discuss two of them while drawing relevant remarks from other researchers. The first result is the convergence of the strategy profile to an equilibrium of the stage game in self-play, and this is probably the most common and widely adopted evaluation criterion in game theory. As an example, it is shown that if fictitious play converges to a pure strategy profile, then it must converge to a Nash equilibrium [3]. Moreover, while fictitious play does not converge to a Nash equilibrium in general, it has been shown to converge to an equilibrium in some special games (e.g. zero-sum games). However, Shoham et al. and Fudenberg et al. share the concern about the "default blind adoption of equilibria as the driving concept in complex games" [4, 7]. We especially agree with Fudenberg that it does not make much sense to make convergence to equilibrium the main factor used to justify interest in a given learning rule [4]. That is, we question the analysis of convergence behaviour by asking — what happens if a learning algorithm does not converge in self-play? and even if it does, which equilibrium does it converge to when there are many? On the other hand, we recognize that convergence property retains its value in the sense that an equilibrium represents some stable situation in which every agent is best responding. So, this might be more relevant to *mechanism design* particularly if the designer has some ways of providing a driving dynamics that gives rise to the equilibrium — for instance, if the designer has some ways of making agents naturally adopt a learning rule.

The second evaluation criterion is the successful learning of the opponent's strategy. To us, this seems a more sensible criterion from the general sense of learning. Actually, Bayesian learning is shown to converge to the correct model of the opponent's strategy under some conditions. However, these conditions are very strong — so strong that they can hardly be satisfied in general unless agents have enough prior knowledge of the opponent's strategy, which is what they try to learn. Even more unfortunately, Foster et al. prove an *uncertainty principle* which basically says that an rational agent is impossible to learn to predict the opponent's behaviour if the agent is sufficiently ignorant of the opponent's payoff functions [2]. Young then suggests to get around the strongly negative result by relaxing rationality [10]. That is, agents are not perfectly rational and do not (strictly) best respond to their assessed model of the opponent's strategy. Without discussing the details of Young's proposed method, we argue that the concept of relaxing perfect rationality makes sense to us. To address our point, we draw relevance to [9], in which Wright et al show that (Nash) equilibria do not provide a good prediction of actual human play in normal-form games and that other behavioural models giving up perfect rationality can do better at predicting. While this may not appear directly related, we do feel that there is some underlying idea bridging the two discussions.

One final point that we stress is Shoham et al.'s proposed empirical evaluation to complement the formal analysis [7]. Perhaps from more of a computer scientist's perspective, we support that empirical evaluation of learning algorithms be an alternative when formal analysis does not give satisfactory results. While not overlooking the strength of formal analysis, we somehow believe that, like in many aspects of computational sciences, the values of empirical performance may outweigh that of nice formal properties of a learning algorithm in many scenarios in the multi-agent setting.

5 Conclusion

This paper surveys some recent work in MAL and presents a very brief introduction to certain MAL algorithms. It also reviews and discusses the remarks from different researchers about the work in the field. We see MAL as a natural extension of machine learning or single-agent artificial intelligence in general. As such we believe that the field may also benefit from the incorporation with neuroscience and other biological sciences. Meanwhile, we must be very careful in justifying the use of any learning technique in the multi-agent setting and be open to assess its value under various appropriate criteria.

References

[1] E. Alonso, M. d'Inverno, D. Kudenko, M. Luck, and J. Noble. Learning in Multi-Agent Systems. *The Knowledge Engineering Review*, 16(3):277–284, 2001.

[2] D. P. Foster and H. P. Young. On the impossibility of predicting the behavior of rational agents. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 98, pages 12848–12853, 2001.

[3] D. Fudenberg and D. Kreps. Lectures on Learning and Equilibrium in Strategic-Form Games. Technical report, CORE Lecture Series, 1990.

[4] D. Fudenberg and D. K. Levine. An economist's perspective on multi-agent learning. *Artificial Intelligence*, 171(7):378–381, 2007.

[5] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations*. Cambridge University Press, 2009.

[6] Y. Shoham, R. Powers, and T. Grenager. Multi-Agent Reinforcement Learning:a critical survey. Unpublished survey, 2003.

[7] Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007.

[8] C. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.

[9] J. Wright and K. Leyton-Brown. Beyond Equilibrium: Predicting Human Behavior in Normal-Form Games. In *AAAI Conference on Artificial Intelligence*, 2010.

[10] H. P. Young. The possible and the impossible in multi-agent learning. *Artificial Intelligence*, 171(7):429–433, 2007.