Efficient Learning in Games

Raghav Aras, Alain Dutech, François Charpillet

Efficient Learning in Games

Raghav Aras, Alain Dutech and François Charpillet

Maia Team - LORIA/INRIA
Campus Scientifique BP239
54506 Vandoeuvre les Nancy
{aras,dutech,charp}@loria.fr

Abstract : We consider the problem of learning strategy selection in games. The theoretical solution to this problem is a distribution over strategies that responds to a Nash equilibrium of the game. When the payoff function of the game is not known to the participants, such a distribution must be approximated directly through repeated play. Full knowledge of the payoff function, on the other hand, restricts agents to be strictly rational. In this classical approach, agents are bound to a Nash equilibrium, even when a globally better solution is obtainable.
In this paper, we present an algorithm that allows agents to capitalize on their very lack of information about the payoff structure. The principle we propose is that agents resort to the manipulation of their own payoffs, during the course of learning, to find a "game" that gives them a higher payoff than when no manipulation occurs. In essence, the payoffs are considered an extension of the strategy set. At all times, agents remain rational vis-à-vis the information available. In self-play, the algorithm affords a globally efficient payoff (if it exists).

Mots-clés : Reinforcement Learning, Multi-agent System, Game Theory

1 Introduction

A game is a multi-agent decision-making problem where the participating agents may have conflicting interests. The agents' fates are tied together in that the payoff obtained by any agent is decided by the combination of decisions that the agents take. No agent can act independently towards his own interest.
Briefly, a game G is described by the parameters $< N, (A^i)_{i \in N}, (U^i)_{i \in N} >$ (Osborne & Rubinstein, 1994), where N is a set of agents, A^i is agent i's strategy set, and U^i is agent i's payoff function. The set of the combinations in which agents can choose strategies is denoted by A. $U^i(a)$ is the payoff agent i obtains when some combination of strategies $a \in A$ is chosen.
A multi-agent learning problem arises when the agents do not know the payoff functions. Each agent begins learning his own strategy selection probability distribution (referred to hereafter and interchangeably, as a mixed strategy or an *action policy*) that maximizes his average limiting payoff. The learning process is considered perpetual. An oft made assumption about agents in game theory is that they are *all* rational. This

	C	D
C	4, 4	0, 5
D	5, 0	1, 1

Table 1: Game 1 : The Prisoners' Dilemma where both player will rationnaly play D whereas both playing C would be more profitable.

is given to mean that if agents are fully exposed to each other's payoff functions, their choice of strategies is circumscribed by the set of Nash (or correlated) equilibria. A Nash equilibrium is a profile of mixed-strategies, $\pi =< \pi^1, \ldots, \pi^N >$, such that the following holds $\forall\ i, \forall\ \sigma^i \neq \pi^i$:

$$u^i(\pi^i, \pi^{-i}) \geq u^i(\sigma^i, \pi^{-i})$$

π^{-i} denotes the mixed-strategy profile that excludes agent i's mixed-strategy π^i. In words, no agent i has an incentive to change his manner of selecting strategies, π^i, *given* that the other agents choose according to π^{-i}. A game may have more than one (mixed)-strategy profiles constituting an equilibrium. An undesirable feature of the Nash equilibrium solution concept (besides the non-uniqueness), is that the payoffs obtained by the agents playing according to a Nash equilibrium could be improved upon (for every agent) by some non-equilibrium policy profile. A policy profile is said to be Pareto-efficient if no other policy profile gives every agent as much payoff while giving atleast one agent a higher payoff. Thus, if $\forall\ \pi$, $\forall\ i$ and for atleast one j, the following holds, then σ is Pareto-efficient:

$$u^i(\sigma) \geq u^i(\pi)$$
$$u^j(\sigma) > u^j(\pi)$$

1.1 Examples

Consider the two games, Game 1 (described in Table 1) and Game 2 (in Table 2). Both are examples where the Nash equilibrium is Pareto-dominated by a non-equilibrium strategy profile. In Game 1 (Prisoners' Dilemma), a pure-strategy Pareto-efficient solution (C, C) is available, giving both agents higher payoff than (D, D). In Game 2, a mixed-strategy in which each agent plays strategies A and B with probability 0.5 each is Pareto-efficient beating the payoffs of the Nash equilibrium (D, D).

1.2 Learning Algorithms

Some of the earliest algorithms proposed for multi-agent learning in games were of the "equilibrium" learners type. The Minimax-Q (Littman, 1994), Nash-Q (Hu & Wellman, 2003), Correlated-Q (Greenwald & Hall, 2003), Adaptive Procedure (Hart & Mas-Colell, 2000) are instances of this kind. The learning agent learns a mixed-strategy that corresponds to an equilibrium of the game. Hence, an implicit cooperativeness (or a common knowledge of rationality) is assumed amongst the agents. The learner is blind

Table 2: Game 2 : In this game, the mixed strategy $(0.5[A] + 0.5[B])$ beats the utility of the Nash equilibrium (D, D).

	A	B	D
A	4, 3	3, 4	0, 5
B	3, 4	4, 3	0, 5
D	5, 0	5, 0	1, 1

to the actual policy used by the opponents. Evidently, an equilibrium learner can perform very poorly against opponents who deliberately play policies that produce a loss for themselves or who have learnt to play some other equilibrium of the game. Thus, the learning agent can be said to be irrational.

An alternative to equilibrium learners is "best-response" learners. Example algorithms of this type are Joint-Action Learners (Claus & Boutilier, 1998), "Bully" (Littman & Stone, 2001), WoLF-PHC (Bowling & Veloso, 2001b) among others. These algorithms ensure that the learner is rational. In self-play, they are also convergent to (obviously) Nash equilibria. In non-self play, a best-response learner can be exploited or fooled by an opponent who explicitly models the learning dynamic of the learner, as shown by the PHC-Exploiter algorithm (Chang & Kaelbling, 2001). To remedy this, algorithms that attempt to achieve "zero-regret" have been proposed such as IGA (Singh *et al.*, 2000) and GIGA-WoLF (Bowling & Veloso, 2001a). The latter is currently the only best-response algorithm which is convergent and with zero-regret in self-play.

1.3 Efficient Learning

A common feature of all the algorithms is self-play. This is an understandable choice for experimental and theoretical analysis given that an algorithm cannot be expected to be rational and convergent for any kind of opponent; in other words, prediction (of others' play) and optimization (in response to it) cannot be simultaneously achieved (Nachbar & Zame, 1996). For example, opponents could choose to play non-stationary policies, which are optimal in some repeated games, in which case no best-response can be theoretically formulated. There have been few attempts made to categorize agents in terms of their learning algorithmic capacities with (Chang & Kaelbling, 2001) being a notable exception.

We restrict attention to self-play. We argue that given self-play and the agents' non-exposure to the payoff functions, it is reasonable to make agents learn Pareto-efficient policies. If agents are identical they could be implanted with an idea of global efficiency if they are guaranteed that deviations (from the agreed learning process) merely result in the (unsatisfactory) Nash equilibrium, and not something even more sub-optimal. The proposal is sound if at any time each agent plays his best-response strategy to the perceived strategies of the other agents. When agents *know* the payoff functions, playing non-equilibrium strategies does not make sense. However, when they do not know these same payoff functions, merely pursuing the learning of equilibrium strategies does not make sense either. There is no prescriptive force behind it.

To summarize, we present an algorithm that in self-play is convergent to policies that are Pareto-efficient. If any agent does not adhere the learning procedure (within certain limits) the learner still learns a policy that forms a best-response. We also assume that each agent is privy only to the strategy he chooses and the payoff he receives. Choices made by others and payoffs received by them are not visible to the agent. We next present the principle that the algorithm uses. As a reminder, we state that the learning process is perpetual, i.e., the game is an infinitely repeated one. Moreover, using the taxonomy of (Chang & Kaelbling, 2001), the agents are in the league $H_\infty \times B_0$, i.e., they are memory-less. Here, H_l represents an agent who can recall strategies he played in the previous l rounds, while B_l represents the agent's belief that his opponents can recall strategies they played in the last l rounds.

2 Payoff Manipulation

The principle we propose for agents to learn Pareto-efficient strategies, is that of payoff manipulation. Any learning is guided by the payoffs agents receive. Instead of assimilating the payoff directly into an update rule, an agent can make several hypothesis about the numerical value of the payoff. In particular, he can choose to assimilate only a percent of the payoff or to inject a higher payoff in the update rule. In game theory, it is often inquired as to what side-payment an agent can make to another to incite the latter to play a strategy favorable to the former. We consider agents to be isolated from each other, so side-payments cannot be materially made. But the essence of the transaction can be retained if an agent makes the negative of the side-payment to himself that he would have made to the other agent given that payoffs re normalized to a certain standard.

Briefly, we consider that an agent must also learn a *payment* policy. A payment policy for agent i, denoted by p^i, is a function, $p^i : A^i \rightarrow \Re$. Thus, the agent adds $p^i(x)$ to every payoff he receives when he plays the strategy $x \in A^i$. We consider, for this paper, only discrete valued self payments. Hence, $p^i : A^i \rightarrow R^i$, where $R^i = \{-R_{max}, \ldots, -1, 0, 1, 2 \ldots, R_{max}\}$. A payment policy has an entry for every strategy. Thus it is a $|A^i|$ sized vector. Learning a payment policy entails learning an $|A^i|$-sized combination. The payment policy where $\forall\, x \in A^i$, $p^i(x) = 0$ is the *default* payment policy. We only consider *pure* payment policies (A probability distribution over payment policies is a mixed payment policy).

2.1 Transformed Games

Given a profile of payment policies $P =< p^1, \ldots, p^N >$, a game G is transformed to the game G_P as far as the learning process is concerned. An appropriate profile P can result in a game G_P in which the solution that was Pareto-efficient, but irrational in G, becomes rational. That is, it develops into an equilibrium that is Pareto-efficient. In conjunction with this, for a given payment policy, we use an algorithm that solves the learning problem, such as WoLF-PHC, for the modified game.

	A	B	D
A	4, 3	3, 4	0, 3
B	3, 4	4, 3	0, 3
D	3, 0	3, 0	-1, -1

Table 3: Game 2_P : both player have applied a payment policy of $(0, 0, -2)$, then the Nash equilibrium $(0.5[A] + 0.5[B]; 0.5[A] + 0.5[B])$ is Pareto efficient if the orginal Game (of Table 2).

Our goal is to build an algorithm such that converges to the choice of a payment policy that corresponds to a Pareto-efficient solution of the original game.

Fact 1
For every game G, there exists a profile of pure payment policies P, such that the Pareto-efficient solution of G, denoted by σ_G is a (mixed) Nash equilibrium of the game G_P.

Fact 1 indicates that while the Pareto-efficient solution itself might be a mixed-strategy profile, the agents' payment policies can be pure. This encourages Q-learning in the space of "pure" payment strategies as against mixed ones as shown in the next section.

2.2 Example

Consider Game 2. If agents learn in isolation (not seeing each others' strategy choices/payoffs), they cannot collude through the learning process to arrive at the Pareto-efficient solution, which is, for each i, $\pi^i = (0.5, 0.5, 0)$ (probabilities for playing A, B and D respectively). This gives each agent a payoff of 3.5, higher than 1 obtained by playing $(0, 0, 1)$, the Nash equilibrium. The payment policies, $p^r = (0, 0, -2)$ (for the row agent) and $p^c = (0, 0, -2)$ (for the column agent) transform the game to Game 2_P. The numbers in the parenthesis are the side-payments for strategies A, B and D respectively. In this game, the mixed-strategy given earlier, (π^1, π^2) is a Nash equilibrium.
In general, discrete-valued payments will not be adequate.

2.3 Q-values with Payments

A strategy's Q-value is the stochastic approximation of its average payoff. Learning algorithms are based on the online stochastic approximation of these Q-values using the following update rule:

$$Q(a^i) \leftarrow (1 - \alpha)Q(a^i) + \alpha\, U^i(a^i, a^{-i}) \tag{1}$$

When an agent can manipulate the payoffs he receives by adding payments to payoffs, he needs to make an approximation of the average payoff for each strategy-payment pair, using $U^i = u^i + p^i$. Considering Q-values of each pair (a^i, r^i) globally, where $a^i \in A^i$ and $r^i \in R^i$ is not feasible since, as other agents can change their payment

policies, payoffs for agent i will be non-stationary.
Hence, we consider the following approach: each payment policy is itself treated as a strategy. Learning is thus a two-shell procedure. The stochastic approximation of the average payoff of a (fixed) payment policy occurs in the outer shell. This loop runs for a very high number of iterations. In each such iteration (inner shell), a best-response algorithm such as WoLF-PHC is run for a very small number of iterations, resulting in a mixed-strategy equilibrium of the agents' strategies from the set A. The inner shell thus corresponds to a particular game, where the payoff functions are given by the original payoff functions compounded by the fixed payment policies.

3 Algorithms

We present our algorithm, Pareto-efficient Learning, alongwith the underlying algorithms (PHC, WoLF-PHC) in this section. The policy-hill climbing (PHC) algorithm is a Q-learning type algorithm that generates a best-response mixed-strategy rather than a pure strategy, if the oppenents all uses *stationary* strategies. The strategies's weights are in direct proportion of their respective Q-values. α and δ are learning rates ($\leq$

Algorithm 1 PHC

Input: $\forall\ a \in A^i$, $Q^i(a) \leftarrow 0$, $\pi^i(a) \leftarrow \frac{1}{|A^i|}$

1: Choose strategy a, observe payoff u^i
2: $Q^i(a) \leftarrow$ (1 - α)$Q^i(a)$ + $\alpha\ u^i$
3: **if** $a = \arg\max_{b \in A^i} Q^i(b)$ **then**
4: $\quad \pi^i(a) \leftarrow \pi^i(a) + \delta$
5: **else**
6: $\quad \pi^i(a) \leftarrow \pi^i(a) + \frac{\delta}{|A^i|-1}$
7: **end if**

1). This algorithm, while being rational, is not convergent in self-play when the game has only mixed-strategy equilibria. The improvement of this algorithm, "Win or Learn Fast" (WoLF)-PHC (Bowling & Veloso, 2001b), replaces δ by two learning rates, δ_w and δ_l, with $\delta_l > \delta_w$. Using δ_l or δ_w depends on the fact that the player is actually on a losing or winning trend (as computed by using a windowed mean μ in algorithm 2). The WoLF-PHC algorithm is capable to learning mixed-strategy equilibria in self-play. Moreover, it is convergent to these, unlike PHC.
This is an algorithm from the memory-less ($H_\infty \times B_0$) league of (Chang & Kaelbling, 2001). Agents of this league can be "exploited" by agents from the $H_\infty \times B_t$ league, i.e., those endowed with memory, to play sub-optimal (non-best response) policies. In fact, (Chang & Kaelbling, 2001) also presented an algorithm based on this principle. An important concession they make is that agents are able to monitor the activities of one another (strategy chosen).

Algorithm 2 WoLF-PHC

1: Steps 1, 2 of PHC
2: $\mu^i(a) \leftarrow \mu^i(a) + \frac{|\mu^i(a) - \pi^i(a)|}{t}$
3: **if** $\sum_{b \in A^i} Q^i(b)[\pi^i(b) - \mu^i(b)] \geq 0$ **then**
4: $\quad \delta \leftarrow \delta_w$
5: **else**
6: $\quad \delta \leftarrow \delta_l$
7: **end if**
8: Steps 3-7 of PHC

3.1 Pareto-Efficient Learning (PEL)

In the PEL algorithm, there are, under consideration, $|A^i|^{|R^i|}$ "strategies" (payment policies) for each agent. Recall that a payment policy is considered as a strategy. A Q value for each such strategy is stocked. Agents switch payment policies in cohesion, i.e., after a commonly agreed number of iterations, L. This number is the duration of an inner "loop". The WoLF-PHC algorithm is run for this duration. Γ^i denotes the set of all payment policies of agent i.
For every payment policy, the PEL algorithm maintains two Q-values, which are updated by the standard Q-value update (equation 1). The first Q-value, Q_p uses payoff received compounded with the payment from the fixed policy. The second, Q_{p+} is updated using just the payoff received. The payment policy's average payoff is calculated using the (mixed-)strategy learned using WoLF-PHC and the "true" Q-values given by Q_{p+}, and is stored in its own Q-value, denoted by $\hat{Q}$ (see algorithm 3).

Algorithm 3 Pareto-Efficient Learning

Input: $\hat{Q}^i(p) \leftarrow 0, \forall\, p \in \Gamma^i$
1: Select p (using some ϵ-greedy policy on $\hat{Q}^i$)
2: **for** $t = 1$ to L **do**
3: $\quad$ Choose strategy a, observe payoff u^i
4: $\quad Q_p(a) \leftarrow (1-\alpha)Q_p(a) + \alpha(u^i + p(a))$
5: $\quad Q_{p+}(a) \leftarrow (1-\alpha)Q_{p+}(a) + \alpha(u^i)$
6: $\quad$ Runs steps 2-8 of WoLF-PHC
7: **end for**
8: $U_p^i \leftarrow \sum_{b \in A^i} \pi(b) Q_{p+}(b)$
9: $\hat{Q}^i(p) \leftarrow (1-\alpha)\hat{Q}^i(p) + \alpha U_p^i$

3.2 Analysis

The two important aspects of any game-learning algorithm are convergence and rationality, as stated earlier. The PEL algorithm is guaranteed the two properties on account

of two features:

- Since agents move out of the inner loop (lines 2-7) simultaneously, the PEL effectively runs the WoLF-PHC. Empirically, the WoLF-PHC has been shown to converge to a sample Nash equilibrium (possibly composed of mixed-strategies) of the concerned game (Bowling & Veloso, 2001b).
- Agents use the unadulterated Q-values to evaluate a payment policy (line 8). This ensures that the agents remain rational. If an agent does not adhere to the PEL algorithm, the non-deviating agent will eventually learn the default payment policy (zero payment with every strategy). The mixed-strategy learnt in association with this payment policy will be necessarily a Nash equilibrium of game.

3.3 Experiments

We ran the PEL algorithm on Game 2. This game has the relevant properties. This game has a pure Nash equilibrium (D, D). The Pareto-efficient solution is $\pi^i = (0.5, 0.5, 0)$ for both i. One of the payment policies that transforms Game 2 to a game where (π^1, π^2) is a Nash equilibrium is $(2, 2, 0)$, which reads as every time strategies A and B are chosen, 2 is added to the payoff received and 0 is added to the payoff from choosing D. Other payment policies resulting in a similar transform also exist. They represent the situation in single agent Q-learning where more than one optimal strategies exists. In identical-interest multi-agent Q-learning (Claus & Boutilier, 1998), the dynamic of learning makes the agents pick one particular optimal joint-strategy.
We varied the number of iterations of the inner and the outer loop. For example, the number of iterations of PEL was 100,000, and that of the inner loop was 1000 iterations. This represents a total of 100 million iterations. However, convergence to an optimal payment policy was not achieved using the PEL. By convergence, we mean that the change in the Q-value of the payment policies continued to be appreciable after the iterations of the outer loop. The inner loop presents a major bottleneck. Clearly, 1000 iterations is insufficient for the WoLF-PHC to converge. A compromise has to be made in the ratio of the lengths of the outer loop to the inner. Nevertheless, two encouraging signs were seen: in any run, the average payoff for (one of the) optimal payment policies was indeed higher than the rest of the policies, for both the agents. Secondly, the mixed-strategies associated with the optimal payment policy was converging towards the Nash equilibrium of the transformed game.
Clearly, it is desirable to have a tighter interleaving of the payment policy evaluation and mixed-strategy evaluation. We are currently working on ways of doing so.

4 Conclusion and Future work

We have presented an algorithm for learning mixed-strategies in general-sum games that are Pareto-efficient. In essence, it is a straightforward extension of the WoLF-PHC algorithm. However, the technique of searching in the space of games via the principle of payoff manipulation is novel. Current research in game learning focuses on Nash equilibria, even in self-play. We argue that if agents are identical, they can learn in a

way that permits them to remain “rational” while playing strategies that give them a highest possible payoff.

References

BOWLING M. & VELOSO M. (2001a). Convergence and no-regret in multiagent learning. *Proceedings of the Eighteenth International Conference on Machine Learning*, (17), 27–34.

BOWLING M. & VELOSO M. (2001b). Rational and convergent learning in stochastic games. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, p. 1021–1026.

CHANG Y. & KAELBLING L. (2001). Playing is believing: the role of beliefs in multi-agent learning. *Advances in Neural Information Processing Systems 14*.

CLAUS C. & BOUTILIER C. (1998). The dynamics of reinforcement learing in cooperative multiagent systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*.

GREENWALD A. & HALL K. (2003). Correlated-q learning. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*.

HART S. & MAS-COLELL A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, **68**(5), 1127–1150.

HU J. & WELLMAN M. (2003). Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*.

LITTMAN M. (1994). Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*.

LITTMAN M. & STONE P. (2001). Leading best-response stratgies in repeated games. *Int. Joint Conf. on Artificial Intelligence (IJCAI-2001) workshop on Economic Agents, Models,and Mechanisms*.

NACHBAR J. & ZAME W. (1996). Non-computable strategies and discounted repeated games. *Economic Theory*.

OSBORNE M. & RUBINSTEIN A. (1994). *A course in game theory*. Cambridge, MA: MIT Press.

SINGH S., KEARNS M. & MANSOUR Y. (2000). Nash convergence of gradient dynamics in general-sum games. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*.