Associative Reinforcement Learning
A Proposal to Build Truly Adaptive Agents and Multi-agent Systems

Eduardo Alonso[1] and Esther Mondragón[2]
[1]*Department of Computer Science, City University London, London, EC1V 0HB, U.K.*
[2]*Centre for Computational and Animal Learning Research, St Albans, AL1 1RQ, U.K.*
E.Alonso@city.ac.uk, e.mondragon@cal-r.org

Keywords: Reinforcement Learning, Associative Learning, Agents and Multi-agent Systems.

Abstract: In this position paper we propose to enhance learning algorithms, reinforcement learning in particular, for agents and for multi-agent systems, with the introduction of concepts and mechanisms borrowed from associative learning theory. It is argued that existing algorithms are limited in that they adopt a very restricted view of what “learning” is, partly due to the constraints imposed by the Markov assumption upon which they are built. Interestingly, psychological theories of associative learning account for a wide range of social behaviours, making it an ideal framework to model learning in single agent scenarios as well as in multi-agent domains.

1 INTRODUCTION

Emergent technologies such as the Internet demand personal, continuously running, independent systems. Therefore, for software systems to perform successfully in real-life applications they must be able to behave in an autonomous, flexible manner in unpredictable, dynamic, typically social domains. In other words, new software systems are *agents*.

As any other concept in computer science, defining agency is controversial. A weak notion prescribes autonomy, social ability, reactivity, and pro-activeness, to which a strong one adds various mental states, emotions, and rationality. How these features are reflected in the system’s architecture will depend on the nature of the environment in which the agent is embedded and on the degree of control that the designer has over this environment, the state of the agent, and the effect of its actions on the environment.

It can be said, therefore, that, at first glance, learning does not seem to be an essential part of agency. In fact, agent research has moved from investigating agent components, including learning, to multi-agent systems organization and performance.

Yet, one of the main arguments against considering learning as a requisite for agency is that there are scenarios in which agents can be used and learning is not needed. For example, little can be learned in accessible domains where agents can obtain complete, accurate, up-to-date information about the environment's state, or in deterministic domains where any action has a single guaranteed effect, or in static domains where the environment remains unchanged unless an action is executed.

It is our contention though that in such domains agents are not strictly necessary and that applying object-oriented (OO) technology would suit best the requirements and constraints the designers must meet and abide by. Put roughly, if you can use objects, do not use agents. Unlike agent-oriented technology, OO technology is well established and understood, and enjoys clear modeling and specification languages (UML) and programming languages (Java, C++). On the other hand, as stated in (Alonso, 2002), a Unified Agent Modeling Language is still under development, and although some OO features such as abstraction, inheritance and modularity make it easier to manage increasingly more complex systems, Java (or its distributed extensions JINI and RMI) and other OO programming languages cannot provide a direct solution to agent development.

On the other hand, agents are ideal for uncertain, dynamic systems. The “Laws of Software Evolution”, particularly, those referring to “continuing change” and “increasing complexity” have proven true with the growth of the Internet, and the arrival of cloud computing, and agile computing.

Certainly, it has become increasingly complicated to model and control the way software systems interact and get co-ordinated. Designers cannot foresee in which situations the systems will encounter themselves or with whom they will interact. Consequently, such systems must adapt to and learn from the environment so that they can make their own decisions when information comes. To sum it up, agents need to learn in real-life domains. Therefore, in real-life domains, learning is essential to agency.

We need to be more specific however: Various machine learning methods, notably supervised learning methods, are not easily applied to real-life domains since they typically assume a "teacher" which can provide the agents with the correct behaviour for a given situation. Thus the large majority of the papers in this field have used reward-based methods. In turn, reward-based learning literature may be approximately divided into two subsets (Stone and Veloso, 2000): reinforcement learning methods which estimate value functions; and stochastic search methods such as evolutionary computation, simulated annealing, and stochastic hill-climbing, which directly learn behaviours without appealing to value functions. In this paper we focus on reinforcement learning.

The rest of the paper is structured as follows: In the next two sections, we will survey reinforcement learning techniques applied to single-agent and multi-agent scenarios respectively. Then, our proposal to improve existing reinforcement learning models and algorithms (Q-learning in particular) by incorporating associative principles currently used to explain trial and error learning in animals is introduced. We shall finish with conclusions.

2 SINGLE-AGENT LEARNING

2.1 Reinforcement Learning

Reinforcement learning has been defined as learning what to do – how to map situations to actions – so as to maximise a numerical reward signal.

In its simplest form, the reinforcement learning problem is presented as follows: An agent exists in an environment described by some set of possible states. Each time it performs an action in some state the agent receives a real-valued reward that indicates the immediate value of this state-action transition. This produces a sequence of states, actions, and immediate rewards. The agent's task is to learn an optimal control policy, *i.e.*, a policy that maximizes the expected sum of rewards, with future rewards discounted exponentially by their delay. In other words, the idea is to learn a policy that maximizes the cumulative value for all the states. The learner is not told which actions to take, but instead must discover which actions yield the most reward by exploiting and exploring their relationship with the environment. Typically, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics, trial and error search and delayed reward, are the two most important features of reinforcement learning.

2.2 Techniques

Several techniques have been used to solve this problem, namely: Dynamic Programming, Monte Carlo methods, and Temporal Difference learning (Sutton and Barto, 1998). These techniques work under different assumptions about the model of the environment they use, and about whether or not they bootstrap, that is, whether or not they update estimates based on other learned estimates, without waiting for a final outcome. Dynamic Programming refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment. Because of the unrealistic nature of this assumption, and their great computational expense, classical dynamic programming algorithms are of limited utility. In contrast, Monte Carlo methods require only experience – sample sequences of states, actions, and rewards from on-line or simulated interaction with the environment. Without prior knowledge of the environment's dynamics these methods can still attain optimal behaviour. Finally, Temporal Difference learning is a combination of Monte Carlo ideas and Dynamic Programming ideas. Like Monte Carlo methods, they can learn directly from raw experience. Like Dynamic Programming, Temporal Difference methods do bootstrap.

Temporal Difference methods are the most commonly used reinforcement learning techniques due to their great simplicity. They can be applied on-line to experience generated from interaction with an environment, and they can be expressed nearly completely by simple equations that can be implemented with small computer programs. Allegedly the most popular reinforcement learning algorithm is Q-learning, an off-policy algorithm where the optimal expected long-term return is locally and immediately available for each state-action pair. A one-step-ahead search computes the long-term optimal actions without having to know anything about possible successor states and their values. Under certain assumptions, Q-learning has been shown to converge with probability 1 to the optimal policy.

2.3 Problems

Regardless of their popularity in the machine learning community several difficulties have so far prohibited the application of reinforcement learning techniques to real-life problems:

1. Exploration-exploitation balance: Unlike other machine learning paradigms, reinforcement learning assumes that, for optimal performance, agents explore (state-action pairs for which the outcome is unknown) and exploit (those state-action pairs for which rewards are known to be high). Finding the right balance between exploration and exploitation is not, however, a straightforward exercise;

2. Temporal discounting: A discount factor is set for delayed rewards representing the fact that it is preferred to obtain the reward sooner rather than later. The problem is that small discounts can make the learner too greedy for present rewards and indifferent to the future, while large discounts slow down learning;

3. Generalisation: This approach does not allow for the “transfer” of learning between different yet similar situations. What is learned depends on the reward structure – if the rewards change, learning has to start over;

4. Large state spaces: Despite the apparent success of systems that have incorporated function approximation algorithms that substitute lookup tables, for most practical tasks with large state spaces reinforcement learning fails to converge. Besides, it generates extreme computational costs when not dealing with small numbers of state-action pairs – which are very rare in any real learning scenario. For example, in Q-learning all state-action pairs must be repeatedly visited, which in practice means that many thousands of training iterations are required for convergence in even modest-sized problems.

3 MULTI-AGENT LEARNING

Broadly speaking, multi-agent learning is the application of machine learning techniques to problems involving multiple agents. We focus on a how reinforcement learning may be applied to multi-agent systems.

3.1 The Four Agendas

Four agendas to solve the multi-agent learning problem have been identified (Shoham et al., 2003):

a) The *descriptive agenda* asks how humans learn in a context of other learners;

b) The *distributed AI agenda* focuses on how a central designer controls the way in which learning tasks are decomposed among different agents. Team Learning constitutes a variety of this kind of learning, where a learner discovers a set of behaviours for a team of agents. In this approach, multi-agent learning uses standard single-agent machine learning techniques to maximize global utility;

c) The *equilibrium agenda* studies the problem of multi-agent learning from a game-theoretic perspective. This proposal pivots around the concept of Nash equilibrium: No single agent should have a rational incentive (in terms of a better payoff) to change its individual strategy away from the equilibrium. The theory of learning in games provides the designer with many useful tools for determining the possible equilibrium points of such a system, and has thus been the most popular in the multi-agent learning community;

d) The *AI agenda* adopts the “optimal agent design” perspective and does not consider the equilibrium concept to be central or even necessarily relevant. Instead, single-agent learning where there is only one learner trying to maximise its own utility value is used, and the behaviours are plugged into only one agent rather than distributed amongst multiple agents.

3.2 Reinforcement Learning and the Equilibrium Agenda

Supervised learning methods such as artificial neural networks and pattern recognition are not easily applied to the multi-agent learning since they typically assume a critic which can provide the agents with the correct behaviour for a given situation, an unrealistic assumption when dealing with large collections of independent agents. Thus the large majority of papers in this field have used reward-based methods, reinforcement learning methods in particular, to the extent that the Multi-Agent Learning problem can be re-defined as the Reinforcement Learning problem for Multi-Agent Systems. Different options have been explored:

- The simplest way to extend single-agent Q-learning algorithms to multi-agent Stochastic Games (SG) is just to add a subscript to the original Q formula, that is, to have the learning agent pretend that the environment is passive (*e.g.*, Sen et al., 1994). However simple this technique may be, the definition of the Q-values assumes incorrectly that they are independent of the actions selected by other agents;
- To solve this problem Littman (Littman, 1994) suggested a minimax Q-learning algorithm for zero-sum games (two-person strictly competitive games where what one gains, the other loses). The problem is that minimax-Q is no longer well motivated in general-sum SGs;
- One alternative is to try to explicitly maintain a belief regarding the likelihood of the other agents' policies, and update the value function based on the induced expectation of the Q-values. Claus and Boutilier implemented such an idea with Joint Action Learners in the context of common-payoff games (aka team games or pure co-ordination games) in which agents that receive the same pay-off at each outcome co-operate (Claus and Boutilier, 1998);
- Hu and Wellman proposed Nash-Q learning that updates the values based on some Nash equilibrium on a special class of SGs (Hu and Wellman, 2001). For general-sum games, several refinements of such an algorithm have been implemented, *e.g.*, the Friend-or-Foe algorithm (Littman, 2001), Correlated-Q learning (Greenwald et al., 2002), EXORL (Suematsu and Hayashi, 2002) and Optimal Adaptative Learning (Wang and Sandholm, 2002).

3.3 Problems

However interesting these results are, the fact is that the conditions for convergence are quite restrictive and the results awkward. Nash-Q attempted to treat general-sum SGs, but the convergence results are constrained to the cases that bear strong similarity to the already known cases of zero-sum games and common-payoff games. Furthermore, the constraints they impose are too strong: They must hold for the games defined by the intermediate Q-values throughout the execution of the protocol. It is extremely unlikely that the game will satisfy this condition, and in any case hard to verify at the outset whether it does.

These unsatisfying aspects manifest a deeper set of issues. Regarding the use of Nash equilibrium in the execution of Nash-Q, such equilibrium has no prescriptive force resulting in the existence of multiple equilibria. Bowling and Veloso (Bowling and Veloso, 2001) did spot this problem and put forward two criteria for any learning algorithm in multi-agent settings, namely: (a) Learning should always converge to a stationary policy, and (b) learning should only terminate with a best response to play by the other agent(s). These are useful criteria, but they ignore the fact that one is playing an extended stochastic game. We again confront the centrality of Nash equilibrium to game theory, and whether it should play the same central role in AI.

4 ASSOCIATIVE REINFORCEMENT LEARNING

4.1 Proposal for Single Agents

It has been argued that in order to solve highly complex problems, we must give up *tabula rasa* learning techniques and begin to incorporate psychological bias that will give leverage to the learning process. The necessary bias, we are told, can come in a variety of forms including shaping, local reinforcement signals, imitation, problem decomposition, and reflexes. Indeed, one historical thread of reinforcement learning concerns learning by trial and error, which has its roots in the psychology of animal learning. In particular, the "Law of Effect" includes the two most important aspects of trial and error learning, and hence of reinforcement learning: It is selectional (it involves trying alternative responses and selecting among them on the basis of their consequences) and associative (the response is associated with a particular situation). Unfortunately, such early theories have been proved wrong and, as a consequence, reinforcement learning techniques remain based on outdated principles.

Our main proposal is to improve existing reinforcement learning models and algorithms (Q-learning in particular) by incorporating current associative theories as follows:

1. Reinforcement learning assumes that agents behaviourally "neutral". We propose to introduce drives that will make the agent approach appetitive stimuli and avoid aversive ones. Moreover, exploration itself should be treated as an internal drive, *i.e.*, the agent would tend to explore its environment by default;
2. To endow agents with the ability to form various types of association other than simple stimulus-

response (S-R) associations, a.k.a. habits, upon which reinforcement learning is based:

- Stimulus-stimulus (S-S) associations that would allow the agent to learn about the relationships among the events that compose its environment;
- Response-outcome (R-O) associations that inform the agent that a response will be followed by a particular outcome (goal-directed behaviour);

3. To take into account associative theory's conception of event representation. Reinforcement learning assumes that the events that enter into associations are irreducible entities. In contrast, learning theory maintains that the events that are associated are not unitary, but may be analysed as sets of component elements. Learning about an event is determined either by the summed associative strengths of the elements that comprise it (elemental theories) or by congifural cues;

4. To redefine outcomes as comprising sensorial and motivational elements, subject to the following rules:

- Both motivational and sensorial components of the outcome would be represented in associations involving that outcome;
- Depending on their motivational value, outcomes can be appetitive or aversive. Thus the probability of a response will be increased if it is followed by an appetitive outcome, but will decrease if followed by an aversive outcome;
- The unexpected omission of an event can also enter into associations – this is called inhibitory learning. The omission of an appetitive outcome constitutes an aversive event, and, conversely, the omission of an aversive outcome acts as an appetitive event;
- Neutral stimuli can also gain reward value, by becoming associated with motivationally significant outcomes (second order conditioning);

5. To take into account the fundamental conditions of association formation proposed by associative theory:

- The contiguity of an event and an outcome is necessary but not sufficient for association formation. Relative predictive value (the outcome is not predicted by any other event that is present), surprisingness (the outcome is not fully predicted), and contingency (the probability of the association event-outcome) are fundamental pillars of association formation;
- Reinforcement learning considers outcomes (rewards) as mere values, and fails to integrate them into the association. All current associative theories reject this assumption, because it fails to account for the fact that if a reward ceases to have value, it will no longer support responding.

4.2 Proposal for Multi-agent Systems

Regarding multi-agent learning, our proposal follows the AI agenda and studies multi-agent learning as learning in multi-agent scenarios. These are forms of single-agent learning in multi-agent systems where agents learn from interaction individually and separately. There may be interactions among the agents, but these interactions just provide input, which may be used in the other agents' learning processes. Not the agents but their learning processes are, so to speak, isolated of each other. Each individual learner typically pursues its own learning goal without explicitly taking care of the other agents' learning goals and without being guided by the wish or intention to support the others in achieving their goals. An agent, thus, learn 'as it were alone'.

Communication (not even indirect communication on which pheromone-based learning algorithms rely) or explicit co-ordination is not an issue therefore – co-operation and competition are not tasks to be solved but emergent properties of the environment. Likewise, agents do not have models of other agents' mental states or try to build models of other agents' behaviours.

In such setting, the main criteria to measure an agent's performance is not its ability to converge to an equilibrium in self-play. We ask what the best learning strategy is for a given agent for a fixed class of other agents in the game, that is, how to design an optimal (or at least effective) agent for a given environment. We follow the AI agenda in that we intend to place computational limitations on (the strategy space of) the agents. Such limitations should be given by recent advances in associative learning theories.

Social psychologists have proved that social learning involves not only the use of social information. The effects of direct experience and the similarities between social learning and classical S-S conditioning are considered as crucial in understanding social behaviour (Griffin, 2004). For example, in the process of predator avoidance acquisition, the predatory cue is considered a conditional stimulus to which observers acquire avoidance responses after the stimulus has been presented in contiguity with an alarmed demonstrator, the unconditioned stimulus. More importantly, there are properties of socially acquired predator avoidance (*e.g.*, the intensity of the unconditioned response increases with that of the unconditioned stimulus, and the fact that there is preferential learning about particular types of stimuli) that provide support of the idea that socially acquired behaviours are mediated by individual learning processes and not by independent social learning mechanisms.

This line of research is complementary to the work done in imitation in the Artificial Intelligence community. Such approach has used social learning theories from psychology to develop adaptive agents that learn from others by observing their behaviour. In particular, (Mataric, 1994) has used vicarious reinforcement to deal with the Credit Assignment Problem.

5 CONCLUSIONS

It is our contention that the proposal outlined in this position paper will strengthen the connection between the study of computational and biological systems. In particular, the approach we advocate will contribute to answering the question of how psychological concepts such as motivation, attention and intention can be modelled in artificial organisms to affect adaptive behavioural modifications and control.

Reinforcement learning algorithms have successfully been applied to simple domains in areas such as navigation robotics, manufacturing, and process control. More powerful algorithms will, no doubt, benefit larger scenarios in industrial applications such as telecommunications systems, air traffic control, traffic and transportation management, information filtering and gathering, electronic commerce, business process management, entertainment, and medical care.

REFERENCES

Alonso, E., (2002). AI and Agents: State of the Art. *AI Magazine*, *23,* 25-29.

Bowling, M. and Veloso, M., (2001), Rational and convergent learning in stochastic games. *In Proceedings of the Seventeenth International Joint Conference on Artificial Inelligence* (IJCAI-2001), pages 1021-1026, Seattle, WA.

Claus, C. and Boutilier, C., (1998), The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746-752.

Greenwald, A., Hall, K., and Serrano, R., (2002), Correlated-Q learning. In *NIPS Workshop on Multiagent Learning*.

Griffin, A. S., (2004), Social learning about predators: A review and prospectus. *Learning & Behavior* 32(1), 131-140.

Hu, J. and Wellman, M., (2001), Learning about other agents in a dynamic multiagent system. *Journal of Cognitive Systems Research*, 2:67-79.

Littman, M. L., (2001), Friend-or-foe Q-learning in general-sum games. In *Proceedings of the Eighteen International Conference on Machine Learning.*

Littman, M. L., (1994), Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, 157-163.

Mataric, M., (1994), Learning to behave socially. In *Proceedings of the Third International Conference on Simulation and Adaptive Behavior*.

Sen , S., Sekaran, M., and Hale, J., (1994), Learning to co-ordinate without sharing information. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 426-431, Seattle, WA.

Shoham, Y., Powers, R., and Grenager, T., (2003*), Multi-Agent Reinforcement Learning: a critical survey.* Technical Report.

Stone, P. and Veloso, M., (2000), Multiagent Systems: A Survey from a Machine Learning Perspective, *Autonomous Robots*, Volume 8(3), 345-383.

Suematsu, N. and Hayashi, A., (2002), A multiagent reinforcement learning algorithm using extended optimal response. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (IJCAI-97), 370-377.

Sutton, R. S. and Barto, A. G., (1998*), Reinforcement Learning: An Introduction*, Cambridge, MA: The MIT Press.

Wang, X. and Sandholm, T., (2002), Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In *Advances in Neural Information Processing Systems* (NIPS-2002).