Proceedings of the International Joint Conference on Information Sciences, pp. 1549-1552, 2003

Target Sets: A Tool for Understanding and Predicting the Behavior of Interacting Q-learners

Nancy Fulda and Dan Ventura
fulda@byu.edu, ventura@cs.byu.edu
Computer Science Department
Brigham Young University

June 10, 2003

Abstract

Reinforcement learning agents that interact in a common environment frequently affect each others' perceived transition and reward distributions. This can result in convergence of the agents to a sub-optimal equilibrium or even to a solution that is not an equilibrium at all. Several modifications to the Q-learning algorithm have been proposed which enable agents to converge to optimal equilibria under specified conditions. This paper presents the concept of *target sets* as an aid to understanding why these modifications have been successful and as a tool to assist in the development of new modifications which are applicable in a wider range of situations.

1. Introduction

Reinforcement learning is a sub-field of machine learning in which the agent is provided with numerical feedback for executing input-output pairs rather than being provided with a set of training examples and corresponding correct outputs. Reinforcement learning can be advantageous because it is uneccessary to compile a set of example situations and correct behaviors ahead of time. Any goal-oriented heuristic which can be calculated online is suitable for training the agent.

Within the field of reinforcement learning, much attention has been given to the Q-learning algorithm [9]. Q-learning is attractive for experimental and analytical purposes because of its relative simplicity, its widespread use, and its convergence guarantees [8]. Successful applications of Q-learning and related algorithms to control tasks include network load balancing [6], block-pushing [7], simulated submersible vehicle navigation [2], and target-following [1].

Unfortunately, straightforward applications of Q-learning to the multiagent domain do not always yield effective results. This may happen for two reasons. First, the convergence guarantees of the Q-learning algorithm frequently become invalid in multiagent environments. Second, even when the Q-learners converge properly, the combination of their final policies may still turn out to be an undesirable system behavior.

These difficulties have motivated several modifications of the Q-learning algorithm for multiagent environments, but a formal explanation of their causes and how they may be reliably prevented is still lacking. This paper moves in that direction by presenting the concept of target sets. Target sets are used to identify three potential difficulties in multiagent reinforcement learning and to show that if all three difficulties are avoided, an optimal system behavior will result. These three difficulties are conflict of interest, action shadowing, and joint action prediction.

This paper then uses the presented concepts to discuss the work of several researchers on Q-learning in multiagent environments. It is shown that each of the modified algorithms has addressed all three of the difficulties described above.

2. Q-learning

A Q-learning agent incrementally learns a set of utilities, called Q-values, which represent the expected time-discounted reward received for performing a specific action in a given state. At each time step, the agent executes an action a_t and receives a reward $r(s_t, a_t)$, where s_t is the current state. The corresponding Q-value is then updated according to the function

$$\Delta Q(s_t, a_t) = \alpha[r(s_t, a_t) + \gamma max_a\{Q(s_{t+1}, a)\} - Q(s_t, a_t)]$$

where $\alpha \in [0, 1)$ is the learning rate and $\gamma \in [0, 1)$ is the discount factor.

Under specified conditions, Q-learning is guaranteed to converge to a set of optimal Q-values

$$Q^*(s,a) = r(s,a) + \sum_t \sum_{s_t} \gamma^t p(s_t|s_{t-1}, a_{t-1}) r(s_t, \pi^*(s_t))$$

where $t = \{1, 2, ...\infty\}$, and $p(s_t|s_{t-1}, a_{t-1})$ is the probability of transitioning to state s_t given the previous state and action. The conditions for convergence include requirements that all state-action transitions be visited infinitely often and that the system be first-order Markovian [8].

The agent's goal is to learn which state-action pairs will maximize some evaluation function. The agent's optimal policy can thus be described as $\pi^*(s) = argmax_a\{eval(s,a)\}$. In typical Q-learning, $eval(s,a) = Q^*(s,a)$, but several modified Q-learning algorithms substitute other evaluation functions. Littman has proposed letting $eval(s,a)$ be a minimax function over the possible actions of both agents in a two-player game [5], while Hu and Wellman suggest an evaluation function which favors Nash equilibria [3].

In the multiagent environment, it is sometimes useful to describe potential system behaviors in terms of the average expected reward that a particular behavior will provide to a given agent. Accordingly, we describe a joint state $\mathbf{s}$ indexed by the individual state s_i of each agent i, and a joint action $\mathbf{a}$ indexed by the agents' individual actions a_i.

In multiagent environments, it is frequently assumed that each agent can perceive the actions taken by the other members of the system. In this case, the agent can also employ an evaluation function based on joint actions, $eval(s_i, \mathbf{a})$. Even when the agents cannot perceive each others' actions, it is often useful to use $eval(s_i, \mathbf{a})$ to describe the preferences an agent would have if it could perceive the joint action space.

3. Optimal Equilibria

In this paper, an optimal equilibrium is defined to be a joint action which is (1) a Nash equilibrium, and (2) pareto-optimal with respect to all other Nash equilibria. This provides a solution concept which simultaneously emphasizes stability and global benefit.

4. Target Sets

The concept of a target set is best described as the set of joint actions which maximize agent i's payoff that are made possible by agent i's optimal policy. More formally,

Definition 4.1 *Let $a_i^* = \pi_i^*(s_i)$ be the individually optimal action selection for agent i in joint state $\mathbf{s}$. Then the* target set $T_i(\mathbf{s})$ *for agent i in state $\mathbf{s}$ is defined as the set of all joint actions $\mathbf{a}^T$ such that $a_i^T = a_i^*$ and $\forall \mathbf{a}$, $eval(s_i, \mathbf{a}^T) \geq eval(s_i, \mathbf{a})$.*

It should be noted that the target sets of all agents in the system can intersect in at most one joint action in any state, and that if the intersection of the target sets is non-empty, its sole member can be shown to be an optimal equilibrium.

Theorem 4.1 *In a Q-learning multiagent system, a joint action $\mathbf{a}|\mathbf{a} \in \cap_{i=1}^{n} T_i(\mathbf{s})$ is an optimal equilibrium for state $\mathbf{s}$.*

Proof 4.1 *Let $\mathbf{a}^T$ be a joint action such that $\mathbf{a}^T \in \cap_{i=1}^{n} T_i(\mathbf{s})$ for some state $\mathbf{s}$. Then by Definition 4.1 $\forall i, \forall \mathbf{a}$, $eval(s_i, \mathbf{a}^T) \geq eval(s_i, \mathbf{a})$. If this is the case, then $\mathbf{a}^T$ must be a Nash equilibrium, because no agent has incentive to change its action selection. $\mathbf{a}^T$ must also be a Pareto-optimal solution, because no joint action exists which increases the evaluation function for any agent. Thus $\mathbf{a}^T$ must be an optimal equilibrium.*

It follows that a system which guarantees a nonempty intersection of target sets will exhibit optimal behavior as long the each agent converges to its optimal policy. It is therefore useful to note the types of situations which can cause $\cap_{i=1}^{n} T_i(\mathbf{s}) = \emptyset$. We briefly consider three such situations.

Definition 4.2 *A* conflict of interest *occurs whenever $\nexists \mathbf{a}^*$ such that $\forall i, \forall \mathbf{a}$, $eval(s_i, \mathbf{a}^*) \geq eval(s_i, \mathbf{a})$.*

Definition 4.3 *An* action shadowing problem *occurs for agent i whenever there exists some individual action $a_i^\dagger$ such that $\forall a_i \neq a_i^\dagger$, $eval(s_i, a_i^\dagger) > eval(s_i, a_i)$, but there does not exist a joint action $\mathbf{a}^\dagger$ (where $a_i^\dagger \in \mathbf{a}^\dagger$) such that $\forall \mathbf{a}$, $eval(s_i, \mathbf{a}^\dagger) \geq eval(s_i, \mathbf{a})$.*

Definition 4.4 *A* joint action prediction problem *occurs whenever there exist joint actions $\mathbf{a}^1, ..., \mathbf{a}^n$, $n \geq 2$, such that $\forall i, \forall j$ $eval(s_i, \mathbf{a}^j) \geq eval(s_i, \mathbf{a}) \forall \mathbf{a}$.*

A conflict of interest causes an empty target set intersection because an agent's target set contains only actions which maximize its evaluation function. When a conflict of interest occurs, there is no joint action that simultaneously maximizes the evaluation function of all agents. Thus, there is no joint action that is simultaneously a member of all target sets.

Action shadowing also causes an empty target set intersection, but for different reasons. Action shadowing occurs because multiple joint actions are aliased to a single action selection of the agent. Thus, the individual action with the highest Q-value may not necessarily correspond to a joint action that maximizes the agent's joint evaluation function. If this happens, then that agent's target set will be empty, and if one agent's target set is empty, then the intersection of all target sets must also be empty.

Unlike conflicts of interest and action shadowing, a joint action prediction problem does not always result in an empty target set intersection. Nevertheless, the existence of more than one potentially optimal equilibrium requires that the agents somehow cooperate with each other in selecting one of these potential equilibria. If the agents fail to coordinate their actions, an empty target set intersection will result.

It can be shown that in the absence of these three problems, the combination of the agents' individually optimal policies will result in an optimal equilibrium.

Theorem 4.2 *If there exists one and only one* $\mathbf{a}^*$ *such that* $\forall i, \forall \mathbf{a}, eval(s_i, \mathbf{a}^*) \geq eval(s_i, \mathbf{a})$ *and* $eval(s_i, a_i^*) > eval(s_i, a_i) \forall a_i \neq a_i^*$*, then* $\mathbf{a}^*$ *is an optimal equilibrium.*

Proof 4.2
When executing its optimal policy, each agent will select an action $\pi_i^*(s_i) = argmax_a\{eval(s_i, a)\}$*. For each agent, this action must be* a_i^**, since we know that* $eval(s_i, a_i^*) > eval(s_i, a_i) \forall a_i \neq a_i^*$*. But this means that* $\forall i, a_i^* \in T_i(\mathbf{s})$*, since* $eval(s_i, \mathbf{a}^*) \geq eval(s_i, \mathbf{a}^*) \forall \mathbf{a}$*. Hence,* $\mathbf{a}^* \in \cap_{i=1}^{n} T_i(s_i)$*, and by Theorem 4.1* $\mathbf{a}^*$ *is an optimal equilibrium.*

5. Modified Q-learning Algorithms

Several researchers have proposed modifications to the Q-learning algorithm in order to encourage optimal system behavior. This analysis shows how some of these algorithms address conflict of interest, the action shadowing problem, and the joint action prediction problem.

5.1. Minimax Q

Michael Littman [5] proposed a variation on Q-learning suitable for two-player zero-sum games. Littman's approach, called Minimax-Q, was characterized by two distinct changes:

- The agent could perceive the actions of its opponent, and maintained a table of joint Q-values.
- The agent used a minimax evaluation criteria over the Q-values, rather than just taking a max.

When played against itself, the Minimax-Q algorithm converged to a classic minimax solution. This solution constituted an optimal equilibrium for the following reasons: (1) Because all minimax solutions are also Nash equilibria, the learned solution was a Nash equilibrium; (2) Because all possible joint actions in a zero-sum game are pareto-optimal, the learned solution was pareto-optimal.

Littman's approach addresses the conflict of interest inherent in zero-sum games by changing the evaluation function of the Q-learner to a minimax operator. With this alteration, conflict of interest is no longer an issue, as every zero-sum game has a minimax solution; hence, a solution that maximizes the evaluation function for both agents.

Minimax-Q addresses the action shadowing problem by having the agents learn joint Q-values rather than individual Q-values. This eliminates the aliasing which causes action shadowing to begin with, and ensures that all agents are able to distinctly identify which individual actions might lead to optimal solutions.

The joint action prediction problem was not explicitly addressed in Littman's algorithm. The relative success of the algorithm indicates that if multiple optimal equilibria exist in Littman's soccer simulation, the agents tend to settle on compatible policies through trial and error.

5.2. Nash Q-learning

Hu and Wellman [3] present the concept of Nash Q-learning, or NashQ, for general-sum games. Like Minimax-Q, each NashQ agent maintains a table of joint Q-values. Each agent also maintains a table of joint Q-values for every other agent in the system, updated based on the action selections and the rewards received by the other agents.

In NashQ, Q-values are updated based on the rewards the agent would receive if all agents maintained a Nash equilibrium from the current state onward. This constitutes a change in the agent's evaluation function. The algorithm is guaranteed to converge to optimal Nash Q-values (and hence, a Nash equilibrium) given that a unique Nash equilibrium exists in every state.

Nash Q-learning addresses conflict of interest by redefining the evaluation function to seek actions which contribute to a Nash equilibrium. Since the algorithm's convergence guarantees require a unique Nash equilibrium in every state, it follows that a mutually preferable joint action must exist for all agents in every state.

Like Minimax-Q, NashQ addresses the action shadowing problem by allowing the agents perceive the joint action space and maintain joint Q-values.

Nash Q-learning addresses the joint action prediction problem through the constraints on the convergence proof. Since each state is constrained to contain only one Nash equilibrium, a joint action prediction problem can never occur in the controlled case.

Hu and Wellman have applied their algorithm to less constrained domains and have observed that NashQ converges correctly in many of them, even without the constraints required by the convergence proof. This demonstrates that while a joint action prediction problem may cause an empty intersection of target sets, it does not always do so.

5.3. Optimistic Updating

Lauer and Riedmiller [4] propose a modified Q-learning algorithm that provably finds optimal solutions for cooperative systems in deterministic environments. The algorithm is based on an optimistic updating technique in which each agent assumes that the maximum observed reward is always attainable. The Q-value update equation becomes

$$Q(s_t, a_t) = max\{Q(s_t, a_t), r(s_t, a_t)+\gamma max_a\{Q(s_{t+1}, a)\}\}$$

In the event that two Q-values are equal, Lauer and Riedmiller's algorithm institutes a coordination mechanism in which the Q-value which was first updated to the current value is taken as the agent's policy.

This simple yet effective algorithm addresses conflict of interest by constraining the system to be cooperative. Because the agents share the same utility values, any joint action that maximizes the evaluation function for one agent maximizes it for all the others as well.

With regard to action shadowing, Lauer and Riedmiller's optimistic updating strategy has the same effect as learning Q-values for the entire joint action space. This overcomes the action shadowing problem by allowing the agents to easily identify individual actions that will maximize the evaluation function. Their approach demonstrates that the action shadowing problem can sometimes be addressed without learning joint Q-values. This is advantageous because the size of the joint action space grows exponentially with the number of agents.

Lauer and Riedmiller's coordination mechanism addresses the joint action prediction problem by implementing a tie-breaking procedure. In the event of multiple optimal equilibria, the agents are constrained to select individual actions which allow the first such equilibrium encountered to be executed.

6. Conclusion

Conflict of interest, the action shadowing problem, and the joint action prediction problem can all result in an empty target set intersection and consequential sub-optimal behavior for a multiagent Q-learning system. This paper has shown that in the absence of these three problems, convergence to an optimal equilibrium can be guaranteed. The paper has also shown how modifications to the Q-learning algorithm have obtained better results by addressing each of these three issues. The concept of target sets is therefore a useful tool in analyzing multiagent Q-learning systems and in understanding why they behave as they do. This concept may also help to motivate the development of further algorithms which reliably converge to optimal equilibria.

References

[1] M. Carreras, P. Ridao, J. Batlle, T. Nicosebici, and Z. Ursulovici. Learning reactive robot behaviors with neural q-learning. In *IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics*, Cluj-Napoca, Romania, 2002.

[2] C. Gaskett, D. Wettergreen, and A. Zelinsky. Q-learning in continuous state and action spaces. In *Proceedings of the 12th Australian Joint Conference on Artificial Intelligence*, Sydney, Australia, 1999.

[3] J. Hu and M. Wellman. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, to appear, 2003.

[4] Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the 17th International Conference on Machine Learning*, pages 535–542, San Francisco, CA, 2000. Morgan Kaufman.

[5] Michael Littman. Markov games as a framework for multiagent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, 1994.

[6] Andrea Schaerf, Yoav Shoham, and Moshe Tennenholtz. Adaptive load balancing: A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 2:475–500, 1995.

[7] Sandip Sen, Mahendra Sekaran, and John Hale. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 426–431, 1994.

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

[9] C. J. C. H. Watkins. *Learning from Delayed Rewards.* PhD thesis, University of Cambridge, 1989.