RC22801 (W0305-083) May 16, 2003
Computer Science

Hyper-Q Learning of Mixed Strategies in Multi-Player Normal Form Games

Gerald J. Tesauro
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Hyper-Q Learning of Mixed Strategies in Multi-Player Normal Form Games

Gerald Tesauro TESAURO@WATSON.IBM.COM
IBM Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 USA

Abstract

This paper proposes an extension of Q-Learning, dubbed "Hyper-Q" Learning, which can learn mixed strategies in multi-player normal form matrix or stochastic games. Factors governing the possible convergence of Hyper-Q learning are addressed, including observability of the opponents' mixed strategies. A model-free Bayesian technique is proposed for mixed strategy estimation given the history of observed actions. Hyper-Q is tested in Rock-Paper-Scissors against an Iterated Gradient Ascent (IGA) player, and a Policy Hill Climber (PHC) player. The Hyper-Q learner is able to significantly exploit both of these opponents, and with Bayesian estimation it achieves much better results than with simple Exponential Moving Average estimation.

1. Introduction

Reinforcement Learning (RL) algorithms such as Q-Learning (Watkins, 1989) are advantageous for single-agent learning in a stationary environment, because they can learn on-line without a model of the environment, using only observed rewards and state transitions. A major focus of current research is to extend RL to multi-agent games. The traditional approach to games, computation of Nash equilibrium strategies, is generally of little utility in most practical multi-agent problems. Obstacles to the practical use of game theory include: hidden or imperfect state information, intractability of computing the equilibria, difficulty of equilibrium coordination (collectively agreeing on a choice from amongst many possible equilibria), and the likelihood that other agents may be "irrational," i.e. they may implement something different from a Nash equilibrium strategy.

The application of standard RL algorithms in such games is problematic, however, because: (a) they only learn deterministic policies, whereas mixed strategies are generally needed; (b) the environment is generally non-stationary due to the adaptation of other agents' strategies. In some cases the use of normal single-agent Q-learning in a multi-agent environment may give good empirical results (Kephart & Tesauro, 2000; Sridharan & Tesauro, 2000), but in general one would expect that new algorithms are needed.

A variety of multi-agent extensions of Q-Learning have recently been published. Littman (1994) and (**?**) extend Q-Learning to two-player zero-sum and general-sum games, respectively, and are designed to converge to Nash equilibrium solutions. These algorithms explicitly assume game-theoretic play by the opponent. Littman (2001) and Hall and Greenwald (2001) further consider expansion of the equilibrium solution space to include concepts such as correlated equilibrium. Note that these algorithms all assume full-information games where every agent knows every other agent's payoff. In contrast, Bowling and Veloso (2002) study a Policy Hill Climber (PHC) variant of Q-Learning that only uses its own actions and payoffs. The aim of PHC is to achieve a best-response to whatever strategies are used by other players, regardless of whether or not they are game-theoretic. When combined with the "Win or Learn Fast" (WoLF) principle for dynamically adapting the learning rate, PHC achieves equilibrium convergence in a number of interesting multi-agent scenarios.

This paper proposes an extension of Q-Learning, dubbed "Hyper-Q" Learning, which can learn mixed strategies in multi-player normal form matrix or stochastic games, using observations of other agents' play, but without knowledge of other agents' payoffs. The key idea of Hyper-Q Learning is to learn a value function of state-action pairs, where the "actions" are entire mixed strategies rather than base actions, and the "states" consist of observations or estimates of the opponents' current mixed strategies, plus any additional stochastic-game state description. Given this viewpoint, certain classes of opponent learning rules such as Iterated Gradient Ascent (IGA) (Singh et al., 2000) and Replicator Dynamics (Weibull, 1995) provide stationary "state-transition" rules. In this case it may be possible for the Hyper-Q learner to achieve an exact optimal strategy, provided that the issues of function approximation and strategy observability can be handled successfully.

In cases where the opponent strategy is not directly observable, the paper proposes two model-free techniques for estimating the opponent's current mixed strategy based on recent observed actions: (1) a simple exponential moving average of recent actions; (2) a more sophisticated Bayesian inference technique, in which a probability for every state in the strategy space is estimated, by applying a recency-weighted version of Bayes' rule to the observed action sequence.

The Hyper-Q algorithm is implemented using both of the above state-estimation methods, as well as with perfect state observability, and is tested in a repeated two-player zero-sum game (Rock-Paper-Scissors) against two types of adaptive opponents: an IGA player and a PHC player. The IGA player's adaptation rule is stationary and history-independent, while the PHC player's adaptation does have history dependence. While not having demonstrated exact convergence, we find that the Hyper-Q learning does make significant progress against both of these opponents, as demonstrated by large positive average reward and by reduction in Bellman error as a function of training time.

Section 2 of the paper develops a general formulation of Hyper-Q for stochastic games. Section 3 discusses a number of important issues affecting the possible convergence of Hyper-Q Learning. Section 4 presents the proposed Bayesian state estimation technique, and compares it to a simple Exponential Moving Average scheme. Section 5 discusses implementation details of Hyper-Q in Rock-Paper-Scissors and presents test results against IGA and PHC. Concluding remarks are given in section 6.

2. General Hyper-Q formulation

Recall the formulation of normal Q-learning for a single agent in a finite MDP, consisting of discrete time steps t, a finite state space S, and a finite action set $A(s)$ for every $s \in S$. At each time step t, the agent observes a state s, chooses a legal action a in this state, and then observes a payoff or immediate reward r and a transition to a new state s'. The equation for Q-learning is then given by:

$$\Delta Q(s,a) = \alpha(t)[r + \gamma \max_b Q(s',b) - Q(s,a)] \quad (1)$$

where γ is a discount parameter, and $\alpha(t)$ is an appropriate learning rate schedule. Given a suitable method of exploring all possible state-action pairs, equation 1 is guaranteed to converge to the optimal value function Q^*, and the associated greedy policy is guaranteed to be an optimal policy π^* for the given MDP.

The generalization of a single-agent MDP to multiple agents is called a *stochastic game*, also known as a Markov game. In a stochastic game, at each time step, each agent i independently chooses a legal action a_i in state s. The transition to a new state s' and payoff r_i to agent i are now functions of joint actions of all agents. An important special class of stochastic games are matrix games, in which $|S| = 1$ and payoffs are functions only of joint actions.

Rather than choosing the best action in a given state, an agent's learning or optimization task in a stochastic game is to choose the best *mixed strategy* $\vec{x}_i = \vec{x}_i(s)$ given the known or expected mixed strategy $\vec{x}_{-i}(s)$ of all other agents. Here $\vec{x}_i$ denotes a set a probabilities summing to 1 for selecting each of the $N_i = N_i(s)$ legal actions in state s. The space of possible mixed strategies is a continuous $(N_i - 1)$ dimensional unit simplex, and choosing the best mixed strategy is clearly more complex than choosing the best base action.

We now consider extensions of Q-learning that may be appropriate for stochastic games. Given that the agent needs to learn a mixed strategy, and that this strategy may depend on the mixed strategies of other agents, an obvious idea is to have the Q-function evaluate entire mixed strategies, rather than base actions, and to have the "state" description include an observation or estimate of the other agents' current mixed strategy. This idea forms the basis of the proposed Hyper-Q learning algorithm, which is formulated as follows. For notational simplicity, let x denote the Hyper-Q learner's current mixed strategy, and let y denote an observed or estimated joint mixed strategy of all other agents (hereafter referrred to as "opponents"). At time t, the agent generates a base action according to x, and then observes a payoff r, a new state s', and a new estimated opponent strategy y'. The Hyper-Q function $Q(s,y,x)$ is then adjusted according to:

$$\begin{aligned} \Delta Q(s,y,x) \; = \; & \alpha(t)[r + \gamma \max_{x'} Q(s',y',x') \quad (2) \\ & -Q(s,y,x)] \end{aligned}$$

The greedy policy associated with any Hyper-Q function is thus the mixed strategy x^* defined by:

$$x^* = \arg\max_x Q(s, y, x) \tag{3}$$

3. Convergence of Hyper-Q learning

There are four major issues that must be examined to determine the conditions under which Hyper-Q learning may converge: (1) exploration; (2) function approximation; (3) opponent strategy dynamics; (4) opponent strategy estimation. We consider the first three issues below, followed by a separate discussion of strategy estimation in the following section.

3.1. Function approximation

The proposed Hyper-Q function is a function of both continuous actions (the agent's mixed strategy) as well as continuous state (the opponents' mixed strategies). To represent and learn such functions, one would in general expect that some sort of function approximation scheme, possibly including discretization schemes such as a uniform grid, would be necessary. Establishing convergence of Q-learning with function approximation is substantially more difficult than for a normal Q-table for a finite MDP, and there are a number of well-known counterexamples. One important point to note regarding finite discretization schemes is that they may cause a loss of the Markov property of the underlying MDP (Munos, 1997).

The development of function approximation schemes that enable Q-learning to work well in continuous state spaces and action spaces has been an active research topic in recent years. A number of promising schemes have been published, including locally weighted regression (Smart & Kaelbling, 2000) and tree-based discretization (Uther & Veloso, 1998). There is a least one discretization scheme, called *Finite Difference Reinforcement Learning*, that is provably convergent to the optimal value function of the underlying continuous problem (Munos, 1997).

In the implementation section of this paper, we will study a simple uniform grid discretization of the mixed strategies of the Hyper-Q agent and its opponents. No attempt will be made to prove convergence under this approximation scheme. However, we will argue that for certain types of opponent strategy dynamics described below, a plausible working hypothesis is that a Finite-Difference-RL implementation of Hyper-Q will be provably convergent.

3.2. Exploration

Normal Q-learning requires visiting every state-action pair infinitely often in order to guarantee convergence. For training in a real environment, the issue of how to achieve this is sometimes swept under the rug. The clearest way to achieve this in simulated Q-learning is through the use of *exploring starts* (Sutton & Barto, 1998), in which training consists of a large number of episodes, each of which starts from a randomly selected state-action pair. Exploring starts may not be feasible in a real environment. In this case one may utilize off-policy randomized exploration, e.g., ϵ-greedy policies. Such procedures will ensure that, for all visited states, every action will be tried infinitely often, but they do not guarantee for general MDPs that all states will be visited infinitely often. As a result one would not expect the trained Q function to exactly match the ideal optimal Q^* for the MDP, although the difference in expected payoffs of the respective policies should be vanishingly small.

The above considerations should apply equally to Hyper-Q learning in a stochastic game. The use of exploring starts for states, agent and opponent mixed strategies should guarantee sufficient exploration of the state-action space (assuming finite resolution of whatever function approximator is used). Without exploring starts, the agent can use ϵ-greedy exploration to at least obtain sufficient exploration of its own mixed strategy space. If the opponents also do similar exploration, the situation should be equivalent to normal Q-learning, where some stochastic game states might not be visited infinitely often, but the cost in expected payoff should be vanishingly small. If the opponents do not explore, the effect could be a further reduction in effective state space explored by the Hyper-Q agent (where "effective state" = stochastic game state plus opponent mixed strategy state). Again this should have a negligible effect on the agent's long-run expected payoff relative to the policy that would have been learned with opponent exploration.

3.3. Opponent strategy dynamics

Given that the evolution of opponent mixed strategies over time can be governed by arbitrarily complicated dynamical rules, it seems unlikely that Hyper-Q learning will converge for arbitrary opponent strategy dynamics. Nevertheless, some broad categories of strategy dynamics can be identified under which convergence should be achievable.

One very simple example is that of a fixed opponent mixed strategy, i.e., $y(s)$ is a constant independent of time and independent of Hyper-Q strategy x. Simple examples would be a Rock-Paper-Scissors player that always plays Rock, or that always plays the Nash equilibrium $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. In this case, the stochastic game obviously reduces to an effective MDP with stationary state transitions and stationary payoffs, and with the appropriate conditions on exploration and on learning rates, Hyper-Q will clearly converge to the optimal value function.

Another important broad class of strategy dynamics consists of opponent strategies that evolve according to a fixed, history-independent dynamical rule depending only on themselves and not depending on actions taken by the Hyper-Q player, i.e., $y_{t+1} = f(s, y_t)$. This is a reasonable approximation for many-player games in which any individual player has negligible "market impact," or in which any player's influence on any other player occurs only through a global summarization function (Kearns & Mansour, 2002). In such cases the opponent population strategy representation need not contain details of which player does what, but only needs to express observable summarizations of population activity, such as averages, in order to be acted on by individual agents. An example of such a model is the "Replicator Dynamics" model used in evolutionary game theory (Weibull, 1995), in which a strategy grows or decays in a population according to its fitness relative to the population average fitness. This leads to a history independent first order differential equation $\dot{y} = f(y)$ for the evolution of population average strategy. In these types of models, the stochastic game faced by the Hyper-Q learner again reduces to an effective MDP in which the effective state (s, y) undergoes stationary history-independent transitions, so that Hyper-Q learning should be able to converge.

The final interesting class of opponent strategy dynamics occurs when the opponent can accurately know or estimate the Hyper-Q strategy x, and then adapts its strategy using a fixed history-independent rule: $y_{t+1} = f(s, y_t, x_t)$. This can occur in games where players are required to announce their mixed strategies, or it can occur if the Hyper-Q player voluntarily announces its strategy. An example of this type of model that has been studied for matrix games is the Iterated Gradient Ascent (IGA) model (Singh et al., 2000), in which the agent uses knowledge of the current strategy pair (x, y) to make a small change in its strategy in the direction of the gradient of immediate payoff $P(x, y)$. Once again, this type of model reduces to an MDP with stationary history-independent transitions of effective state depending only on (s, y, x).

Note that the above claims of reduction to an MDP depend in each case on the Hyper-Q learner being able to accurately estimate the opponent mixed strategy y. If this is not possible, then the Hyper-Q learner would instead face a POMDP situation, and the standard Q-learning convergence proofs would not apply. The issue of how opponent strategies may be estimated if they are not directly observable is now taken up in the following section.

4. Opponent strategy estimation

In typical stochastic or matrix games, the opponent mixed strategy is usually not directly observable. Instead, the only observable information available to an agent at time t is the history H_t of joint actions, and the agent's own payoffs. In full-information games the history of opponent payoffs is also observable. In this section we consider techniques for estimating opponent strategies from the history of base actions. One approach to this would be *model-based*, i.e., to consider a class of possible explicit dynamical models of opponent strategy, and then choose the model in the class that best fits the observed data. There are two main problems with this approach: (1) the class of possible models of multi-agent strategy dynamics may need to be extraordinarily large; (2) there is a well-known danger of "infinite regress" of opponent models that can occur when A's model of B attempts to take into account B's model of A.

An alternative approach that we consider in this paper is a *model-free* approach to strategy estimation. This is in keeping with the spirit of Q-learning, which attempts to learn state valuations in an MDP without explicitly modeling the dynamics of the underlying state transitions.

A simple model-free estimation technique used in the following section is the well-known Exponential Moving Average (EMA) technique. This incrementally maintains a moving average $\bar{y}$ of opponent strategy by updates after each observed action according to:

$$\bar{y}(t+1) = (1-\mu)\bar{y}(t) + \mu\vec{u}_a(t) \qquad (4)$$

where $\vec{u}_a(t)$ is a unit vector representation of the base action a (for example, in penny-matching, (1,0) for heads and (0,1) for tails). EMA assumes only that recent observations are more informative about current strategy than older observations, and should give a reasonably accurate estimate when significant strategy changes take place on time scales larger than $1/\mu$.

4.1. Bayesian strategy estimation

A more sophisticated model-free alternative to EMA is now presented. We assume a discretized representation of of the possible values of y, for example, by a uniform grid on the unit simplex representing each individual opponent's mixed strategy. Given the observed history of actions H, a probability value for each discrete y given H, $P(y|H)$, can then be computed using Bayes' rule as follows:

$$P(y|H) = \frac{P(H|y)P(y)}{\sum_{y'} P(H|y')P(y')} \tag{5}$$

where $P(y)$ is the prior probability of state y, and the sum over y' denotes the sum over all discrete points in the strategy space. The conditional probability of the history given the strategy, $P(H|y)$, can now be decomposed into a product of individual action probabilities $\prod_{k=0}^{t} P(a(k)|y(t))$ assuming conditional independence of the individual actions. If all actions in the history are equally informative regardless of age, we may write $P(a(k)|y(t)) = y_{a(k)}(t)$ for all k. This corresponds to a Naive-Bayes equal weighting of all observed actions. However, once again it is reasonable to assume that more recent actions are more informative. The way to implement this in a Bayesian context is with exponent weights w_k that increase with k (Hong et al., 2002). Within a normalization factor, we then write:

$$P(H|y) = \prod_{k=0}^{t} y_{a(k)}^{w_k} \tag{6}$$

An intuitively obvious schedule for the weights is a linear schedule $w_k = 1 - \mu(t-k)$; if the history is truncated at the most recent $1/\mu$ observations, this guarantees that all the weights are positive.

An illustration of the difference between EMA estimation and Bayesian estimation is shown in figure 1. This is taken from a Rock-Paper-Scissors simulation using two IGA players. The curves show : the true probability of player 1 playing Rock, an EMA estimate using equation 4, and a maximum-likehihood Bayes estimate calculated using equations 5 and 6 with a uniform prior. Both estimates use $\mu = 0.005$. The Bayes estimate generally appears to track the true probability better, although the maximum likehihood value is noisier, and in particular, there are large noise spikes at the random strategy restarts occurring every 2000 time steps. This could possibly be addressed by placing greater weight on the prior using a ficititious number of initial observations at the start of a new episode.

5. Implementation and Results

In this section, we examine the performance of Hyper-Q learning in a simple two-player matrix game, Rock-Paper-Scissors. A uniform grid discretization of size $N = 25$ is used to represent the mixed-strategy component probabilities for either player. This implies a simplex grid of size $N(N+1)/2 = 325$ for either player's mixed strategy, and thus the entire Hyper-Q table is of size $(325)^2 = 105625$. All simulations use $\gamma = 0.9$, and for simplicity, a constant learning rate $\alpha = 0.01$.

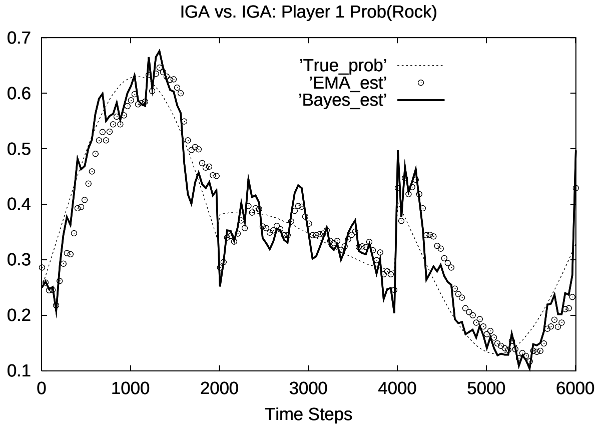

Figure 1. An illustration of estimates of an opponent's mixed strategy based on observations of base actions, taken from a Rock-Paper-Scissors simulation in which both players use the IGA learning procedure, with random strategy restarts occurring every 2000 time steps.

5.1. Hyper-Q/Bayes formulation

Three different opponent state estimation schemes were used with Hyper-Q learning: (1) "Omniscient," i.e. perfect knowledge of the opponent's mixed strategy; (2) EMA, using equation 4 with $\mu = 0.005$; (3) Bayesian, using equations 5 and 6 with $\mu = 0.005$ and a uniform prior.

Modification of equations 2 and 3 were implemented in the Bayesian case to allow for a distribution of possible opponent states y, with probabilities $P(y|H)$. The corresponding equations are:

$$\begin{aligned}\Delta Q(y,x) = \alpha(t)P(y|H)[r + \gamma \max_{x'} Q(y',x') \\ -Q(y,x)]\end{aligned} \tag{7}$$

$$x^* = \arg\max_x \sum_y P(y|H)Q(y,x) \tag{8}$$

A technical note regarding equation 7 is that, to improve tractability of the algorithm, an approximation

$P(y|H) \approx P(y'|H')$ is used, so that the Hyper-Q table updates are performed using the updated distribution $P(y'|H')$.

5.2. Rock-Paper-Scissors results

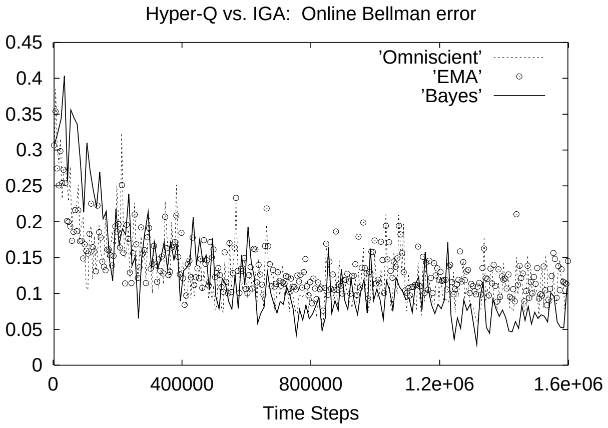

Figure 2. Smoothed online Bellman error for a Hyper-Q learner vs. an IGA player in Rock-Paper-Scissors, using three different opponent state estimation methods: “Omniscient,” “EMA” and “Bayes” as indicated. Random strategy restarts occur every 1000 time steps.

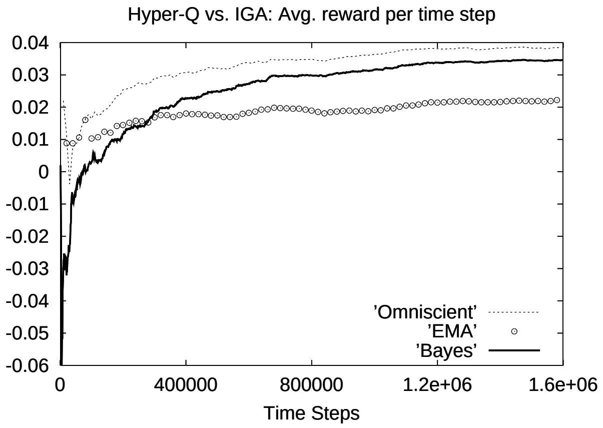

Figure 3. Average reward per time step for a Hyper-Q learner vs. an IGA player in Rock-Paper-Scissors.

We first examine performance of Hyper-Q when training online against an IGA player. As discussed previously, Hyper-Q should in principle be able to converge against this type of opponent, apart from possible state observability and discretization issues. In order to conform to the original implicit assumptions underlying IGA, the IGA player is allowed to have omniscient knowledge of the Hyper-Q player's mixed strategy at each time step. Policies used by both players are always greedy. Restarts every 1000 time steps reset both players' mixed strategies to uniform random values.

Figure 2 shows a smoothed plot of the online Bellman error, while figure 3 shows the Hyper-Q player's average reward per time step, as a function of training time. While not demonstrating exact convergence, the figures do exhibit good progress toward convergence, as suggested by substantially reduced Bellman error and substantial positive average reward per time step. Among the three state estimation techniques used, the Bayesian technique reached the lowest Bellman error at long time scales. This is probably because it updates many elements in the Hyper-Q table per time step, whereas the other techniques only update a single element. Bayes also has by far the worst average reward at the start of learning, but asymptotically it clearly outperforms the EMA technique, and comes close to matching the performance obtained with omniscient knowledge of opponent state.

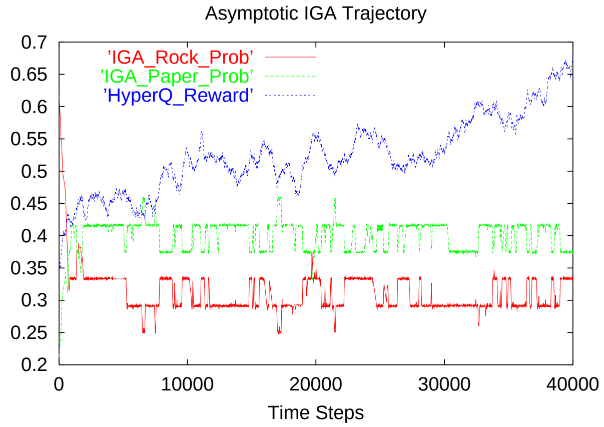

Figure 4. Trajectory of the IGA mixed strategy against the Hyper-Q strategy starting from a single exploring start. Also shown is a suitably rescaled plot of the cumulative reward obtained by the Hyper-Q player.

Part of the advantage obtained by the Hyper-Q player over the IGA player comes from its ability to exploit the transient behavior starting from a random initial condition. In addition, Hyper-Q also appears to exploit the asymptotic behavior of IGA, as shown in figure 4. This plot shows that the initial transient lasts at most a few thousand time steps. Afterwards, the Hyper-Q policy causes IGA to basically cycle, with erratic periodicity, between two different probabilites for Rock and two different probabilities for Paper, thus preventing IGA from reaching the Nash mixed strategy. The overall profit to Hyper-Q during this cycling is positive on average, as shown by the plot of cumulative Hyper-Q reward. The observed cycling with positive profitability is reminiscent of the cycling obtained by an algorithm called PHC-Exploiter (Chang & Kaelbling, 2002) in play against a PHC player. An interesting difference is that PHC-Exploiter uses an explicit model of its opponent's behavior, whereas no such model is needed by a Hyper-Q learner.

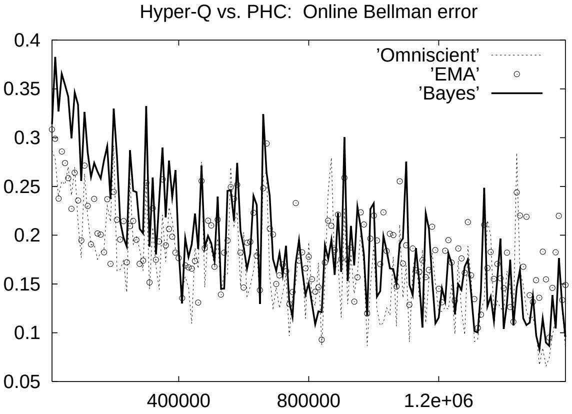

Figure 5. Smoothed online Bellman error for a Hyper-Q learner vs. a PHC player in Rock-Paper-Scissors.

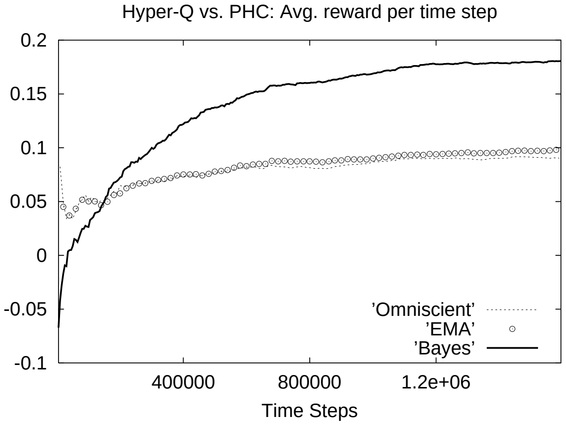

Figure 6. Average reward per time step for a Hyper-Q learner vs. a PHC player in Rock-Paper-Scissors.

We now exmamine Hyper-Q vs. a PHC player. PHC is a simple adaptive strategy based only on its own actions and rewards. It maintains a Q-table of values for each of its base actions, and at every time step, it adjusts its mixed strategy by a small step towards the greedy policy of its current Q-function. The PHC strategy is history-dependent, so that reduction to an MDP is not possible for the Hyper-Q learner. Nevertheless Hyper-Q does exhibit substantial reduction in Bellman error, shown in figure 5, and also significantly exploits PHC in terms of average reward, shown in figure 6. Given that PHC ignores opponent state, it should be a weak competitive player, and in fact it does much worse in average reward than IGA. It is also interesting to note that Bayesian estimation once again clearly outperforms EMA estimation, and surprisingly, it also outperforms omniscient state knowledge. This is not yet well understood and is a focus of ongoing research.

6. Conclusion

This paper has introduced a new multi-agent learning algorithm, Hyper-Q Learning, and a new Bayesian technique for estimating mixed strategies of other agents based on their observed actions. Some tantalizing early results were found in Rock-Paper-Scissors tests against some simple adaptive opponents. These results are extremely recent, and research on this topic should be regarded as very much a work in progress. Obviously there is a need for vastly more research, to develop a satisfactory theoretical analysis of the approach, an understanding of what kinds of realistic environments it can be expcted to do well in, as well as fieldable versions of the algorithm that can be successfully deployed in those environments.

Significant improvements in opponent state estimation should be fairly easy to obtain. The linear formula for setting the recency weights was only a plausible heuristic, and more principled methods should be achievable. For example, Hong et al. (2002) propose a method for training optimal weight values based on observed data. The use of predictive time-series methods such as Kalman filters and ARMA models might also result in substantially better state estimation. Model-based techniques are also likely to be advantageous in situations where one has a reasonable basis for modeling the opponents' dynamical behavior.

It would be interesting to see whether full convergence of Hyper-Q against IGA, possibly employing a Finite-Difference-RL scheme, could be achieved. At a minimum, the use of properly decaying learning-rate schedules $\alpha(t)$ should be investigated. More empirical tests against other types of opponents, where convergence cannot be guaranteed or expected, would also be interesting. One obvious source of opponents is the International RoShamBo Programming Competition (RoShamBo, 2001), a tournament for computerized Rock-Paper-Scissors players.

The most interesting test of Hyper-Q, however, would be to see if it can converge against itself. This has not yet been tried. A Hyper-Q learner certainly is a non-stationary dynamical system when viewed by another Hyper-Q learner, so proving convergence by reduction to an MDP seems unlikely. It is worth noting, however, that if a Hyper-Q learner converges, its dynamics become stationary, so that self-consistent solutions may exist where each agent's Hyper-Q function is optimal given the policy of the other agents. It then becomes an interesting question whether Hyper-Q learning could actually find such solutions when they exist. This is entirely analogous to the situation found for simultaneous Q-learning in two-player dynamic pricing (Kephart & Tesauro, 2000): in some models studied, self-consistent optimal solutions did exist, and in many but not all cases, ordinary Q-learning was able to find them.

Acknowledgements

The author thanks Michael Bowling for helpful discussions regarding implementation of the PHC algorithm, and Irina Rish for helpful discussions regarding Bayesian state estimation.

References

Bowling, M., & Veloso, M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence*, *136*, 215–250.

Chang, Y.-H., & Kaelbling, L. P. (2002). Playing is believing: the role of beliefs in multi-agent learning. *Proceedings of NIPS-2001.* MIT Press.

Hall, K., & Greenwald, A. (2001). Correlated Q-learning. *Proceedings of DIMACS Workshop on Computational Issues in Game Theory and Mechanism Design.*

Hong, S. J., Hosking, J., & Natarajan, R. (2002). *Multiplicative adjustment of class probability: educating naive Bayes* (Technical Report RC-22393). IBM Research.

Hu, J., & Wellman, M. P. (1998). Multiagent reinforcement learning: theoretical framework and an algorithm. *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 242–250). San Francisco: Morgan Kaufmann.

Kearns, M., & Mansour, Y. (2002). Efficient Nash computation in large population games with bounded influence. *Proceedings of UAI-02* (pp. 259–266).

Kephart, J. O., & Tesauro, G. J. (2000). Pseudo-convergent Q-learning by competitive pricebots. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-00)* (pp. 463–470). Morgan Kaufmann.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 157–163). San Francisco: Morgan Kaufmann.

Littman, M. L. (2001). Friend-or-Foe Q-learning in general-sum games. *Proceedings of the Eighteenth International Conference on Machine Learning.* San Francisco: Morgan Kaufmann.

Munos, R. (1997). A convergent reinforcement learning algorithm in the continuous case based on a finite difference method. *Proceedings of IJCAI-97* (pp. 826–831). Morgan Kaufman.

RoShamBo (2001). Second International Roshambo Programming Competition. http://www.cs.ualberta.ca/∼darse/rsbpc.html.

Singh, S., Kearns, M., & Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. *Proceedings of UAI-2000* (pp. 541–548). Morgan Kaufman.

Smart, W. D., & Kaelbling, L. P. (2000). Practical reinforcement learning in continuous spaces. *Proceedings of ICML-00* (pp. 903–910).

Sridharan, M., & Tesauro, G. (2000). Multi-agent Q-learning and regression trees for automated pricing decisions. *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-00)* (pp. 927–934). Morgan Kaufmann.

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

Uther, W. T. B., & Veloso, M. M. (1998). Tree based discretization for continuous state space reinforcement learning. *Proceedings of AAAI-98* (pp. 769–774).

Watkins, C. (1989). *Learning from delayed rewards.* Doctoral dissertation, Cambridge University, Cambridge.

Weibull, J. W. (1995). *Evolutionary Game Theory.* The MIT Press.