Q-learning in Two-Player Two-Action Games

Monica Babes
Rutgers University
babes@cs.rutgers.edu

Michael Wunder
Rutgers University
mwunder@cs.rutgers.edu

Michael Littman
Rutgers University
mlittman@cs.rutgers.edu

ABSTRACT

Q-learning is a simple, powerful algorithm for behavior learning. It was derived in the context of single agent decision making in Markov decision process environments, but its applicability is much broader—in experiments in multiagent environments, Q-learning has also performed well. Our preliminary analysis using dynamical systems finds that Q-learning's indirect control of behavior via estimates of value contributes to its beneficial performance in general-sum 2-player games like the Prisoner's Dilemma.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Algorithms—Intelligent Agents,Multiagent Systems

Keywords

Reinforcement learning, Multiagent learning, dynamical systems

1. INTRODUCTION

Q-learning [14] was one of the first algorithms for reinforcement learning [11] specifically designed to maximize reward in multistage environments. Several authors have shown that it converges to optimal values [12] and optimal behavior [9] in Markov decision process environments.

General sum games provide a formal framework for defining and addressing important problems in agent interaction. Q-learning has been used in multiagent environment and has been shown to perform beyond expectations [8, 6, 2, 7]. In the famous Prisoner's dilemma game, for instance, a player who cooperates gives a payoff of 3 to the other player and 0 to oneself, but defecting gives 0 to the other player and 1 to oneself. Surprisingly, two Q-learning agents repeating this game frequently arrive at an outcome where both choose to cooperate, even when there is no explicit previous history in the state information.

Positive theoretical results have been generalized to some special multiagent environments, in particular when the reward structures are purely cooperative or zero sum [5]. Although Q-learning variants have been defined for general

sum games [4], the very idea of representing multiagent strategies using Q-values has been shown to be extremely problematic [15]. Thus, positive theoretical results have been lacking.

This state of affairs leads us to an enigma. Why is it that Q-learning performs well in multiagent environments even though existing analyses do not support its use in this setting? In this paper, we start collecting clues in the setting of 2-agent, 2-action games where the agents share the same Q-table, which we call *tethered Q-learning*. Tethered Q-learning, while not technically a system of multiple agents, mimics the properties of two separate agents when they start with the same values and learn slowly in the same way. Therefore, it is a good starting point to analyze the dynamics that come from the agents having more freedom in the learning process.

The majority of our results are presented in the context of the Iterated Prisoner's dilemma or IPD [1] because of its simplicity as well as its wide applicability, but still apply to general games. In Section 2, we shed some light on the algorithm's behavior by casting tethered Q-learning into a dynamical system framework and showing that high scoring policies are stable, in some sense. In Section 3, we study tethered Q-learning. We conclude with suggestions for how these insights could be exploited in the design or analysis of future multiagent learning algorithms.

1.1 Definitions

A game is a description of the interaction between two or more agents. We focus on games with two players and two actions, which jointly determine a payoff. In the *iterated* or *repeated games* we study, at each step the environment dispenses rewards to each agent according to the joint action taken by all the agents in the game. A *best response* is a strategy that chooses the action that maximizes the agent's payoff given a fixed specification of the action policies of all the other agents. A *Nash equilibrium* is a joint strategy in which each player's action is a best response to the strategies of its opponents.

Learning algorithms provide a sensible approach for focusing on defining agent behavior in a game from the perspective of a single agent. Many learning algorithms target the generation of a best response, so that no matter what behavior the other agent adopts, the learning agent will be striving to maximize its reward.

1.2 Q-learning

Q-learning is a single-agent learning algorithm that has been used in the multiagent setting. The Q-learner main-

tains a Q-table, which is a data structure that stores a value for each state–action pair. In each state, for each action, the state–action (or Q-) value represents the expected payoff that the agent receives from choosing the given action from that state, then selecting actions in future states to maximize expected payoff.

In a repeated game, the Q-table for each Q-learning agent i consists of a vector of values, Q_i^a, with one component for each action $a \in A$. The Q-learning rule, simplified for this setting, can be written

$$Q_i^a \leftarrow Q_i^a + \alpha(r + \gamma \max_{a' \in A} Q_i^{a'} - Q_i^a),$$

where α is a learning rate or step-size parameter, $0 \le \gamma < 1$ is a discount factor weighting future rewards relative to current rewards, and r is the payoff value received in response to action a. Throughout this paper, we use an ϵ-greedy method to choose between action a that maximizes Q_i^a, and a random action, chosen with probability ϵ.

2. DYNAMICAL SYSTEMS APPROACH

IGA (Infinitesimal Gradient Ascent) was proposed as an abstract algorithm for 2-player, 2-action games [10]. The algorithm maintains an explicit policy for both players, which can be summarized as a single probability for each player specifying its chance of choosing the first action. These policies are updated by taking an infinitesimal step in the direction of the gradient—each player modifies its choices to maximally increase its expected reward. Using a dynamical systems analysis, the authors showed that IGA players converge to a Nash equilibrium or to a repeatedly traversed orbit. In the latter case, the average reward along the loop exactly matches that of a Nash equilibrium. Although IGA is not directly realizable because of the need for arbitrarily small learning rates, it did give rise to several practical algorithms such as WoLF-IGA, GIGA, and GIGA-WoLF [3].

In this section, we view Q-learning in a similar way. We define Infinitesimal Q-learning (IQL), a version of Q-learning where value updates are made infinitely slowly and deterministically—updates based on exploratory actions are made simultaneously. Q-learning was previously analyzed with Boltzmann updating [13]. While seemingly similar (both approaches involve randomly choosing the greedy action with higher probability), there are important differences. The crucial distinction is that when the Q-values for both actions approach parity, Boltzmann chooses each action with close to equal probability while ϵ-greedy jumps between greedy actions creating two distinct regions each with its own greedy action. Whereas IQL with Boltzmann exploration can only converge to mutual best responses (Nash equilibria), we find that IQL with ϵ-greedy includes non-Nash fixed points.

IQL for two-player, two-action games is defined as follows. Let a^* be the action that maximizes the Q-value for player i, $\bar{a}$ be the other action, b^* be the action that maximizes the Q-value for the opponent, and $\bar{b}$ be the opponent's other action. Then,

$$
\begin{aligned}
Q_i^{a^*} \leftarrow Q_i^{a^*} \quad &+\alpha(1-\tfrac{\epsilon}{2})^2 \quad (R_i^{a^*b^*} + \gamma Q_i^{a^*} - Q_i^{a^*}) \\
&+\alpha(1-\tfrac{\epsilon}{2})\tfrac{\epsilon}{2} \quad (R_i^{a^*\bar{b}} + \gamma Q_i^{a^*} - Q_i^{a^*}) \\
Q_i^{\bar{a}} \leftarrow Q_i^{\bar{a}} \quad &+\alpha\tfrac{\epsilon}{2}(1-\tfrac{\epsilon}{2}) \quad (R_i^{\bar{a}b^*} + \gamma Q_i^{a^*} - Q_i^{\bar{a}}) \\
&+\alpha\tfrac{\epsilon^2}{4} \quad (R_i^{\bar{a}\bar{b}} + \gamma Q_i^{a^*} - Q_i^{\bar{a}}).
\end{aligned}
$$

The idea here is that the Q-values, for sufficiently small values of the learning rate α, explore in all directions simultaneously, with the resulting update weighted by its proba-
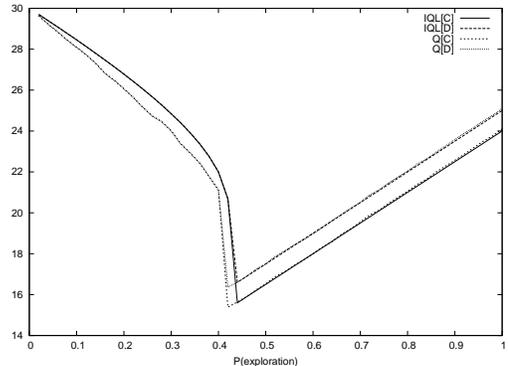
Figure 1: Performance (expected discounted reward) of IQL and Q-learning in IPD for a range of exploration rates.

bility. For example, with the maximum probability $(1-\epsilon)^2$, both players will choose their greedy actions and update their values in the direction of the resulting payoffs. But, with smaller probability, one of the agents will explore, resulting in a different update. The IQL update rule blends these updates together based the exploration-rate parameter ϵ. Note that two IQL agents, starting with identical Q-functions, will remain synchronized as they make the same series of updates. For the remainder of this paper, we completely synchronize the agents' Q-values, so that they share one decision function.

Figure 1 shows the result of IQL in IPD with a range of exploration-rate parameters (discount $\gamma = 0.9$). In each case, starting roughly from mutual cooperation Q-values, IQL converges. As long as exploration stays low, the temptation of defection contributes very little to the update and the higher values of cooperation keep the Q-values high. Once the exploration rate is high enough, though, the values for defection are updated frequently and overtake the values for cooperation. Mutual defection then becomes the only stable solution. Note that, for the low exploration rates, the converged value of cooperation and defection are equal.

3. DIRECT ANALYSIS OF TETHERED IPD

In this section we study tethered Q-learning and calculate the fixed points of the learning algorithm in the space of Q-values. We show that, in the tethered case, the Q-values always converge to some fixed point.

In the tethered case, we distinguish two possibilities in terms of the players' preference for actions: either one of the actions has a higher expected reward and the agents prefer it over the other action, or the two actions have equal values. If the payoff of one of the joint actions dominates all other outcomes for either agent and that action also dominates the other action, then the joint action is a Pareto efficient Nash equilibrium. It will be an attractor for every assignment of Q-values. Similar logic holds when there are two Nash equilibria along the joint action diagonal, but there will be two attractors whose regions depend on the exploration rate as well as the starting points.

In the second case, there may or may not be a Nash equilibrium for joint action (a_1, a_1), but for player i, $R_i^{a_1b_2} > R_i^{a_2b_2}$, and $R_i^{a_2b_2} > R_i^{a_1b_1}$. In this situation, there is some fixed point where the shared Q-values are equal, because

the updates for either greedy action near this point will be pushing the values back towards it. This fixed point will be considered stable. There may also be an unstable point, which features two opposing updates pulling away in opposite directions. In the analysis that follows, we provide a formula for finding these points for general two-player two-action symmetric games, if they exist along the diagonal.

- $R^{a_x b_y}$: average value of reward for row player of action a_x when the other player plays b_y

- R_{xy}: expected rate of reward for action x when x is selected but the greedy action for both players is y

- $Q(a_x^*) = Q_{a_x^*}$: Q-value of greedy action a_x

- $Q(\bar{a_y}) = Q_{\bar{a_y}}$: Q-value of non-greedy action a_y

- $Q_{a_y^*}(t)$: Q-value of greedy action a_y at time t

- t: number of time steps according to Q-learning with learning rate approaching 0, after starting at $t = 0$.

We find these points where the two following equations cancel each other by row to produce zero change in the update. The rewards arrive at the following rates:

$$
\begin{aligned}
R_{11} &= (1 - \frac{\epsilon}{2})[(1 - \frac{\epsilon}{2})R^{a_1 b_1} + \frac{\epsilon}{2}R^{a_1 b_2}] \\
R_{21} &= \frac{\epsilon}{2}[(1 - \frac{\epsilon}{2})R^{a_2 b_1} + \frac{\epsilon}{2}R^{a_2 b_2}] \\
R_{12} &= \frac{\epsilon}{2}[(1 - \frac{\epsilon}{2})R^{a_1 b_2} + \frac{\epsilon}{2}R^{a_1 b_1}] \\
R_{22} &= (1 - \frac{\epsilon}{2})[(1 - \frac{\epsilon}{2})R^{a_2 b_2} + \frac{\epsilon}{2}R^{a_2 b_1}].
\end{aligned}
$$

The first equation updates the greedy action on top, and in the second the greedy action is on the bottom row. [1]

$$
\begin{bmatrix} \frac{\partial Q(a_1^*)}{\partial t} \\ \frac{\partial Q(\bar{a_2})}{\partial t} \end{bmatrix} = \begin{bmatrix} (1 - \frac{\epsilon}{2})(\gamma - 1) & 0 \\ \gamma\frac{\epsilon}{2} & -\frac{\epsilon}{2} \end{bmatrix} \begin{bmatrix} Q(a_1^*) \\ Q(\bar{a_2}) \end{bmatrix} + \begin{bmatrix} R_{11} \\ R_{21} \end{bmatrix}
$$

$$
\begin{bmatrix} \frac{\partial Q(\bar{a_1})}{\partial t} \\ \frac{\partial Q(a_2^*)}{\partial t} \end{bmatrix} = \begin{bmatrix} -\frac{\epsilon}{2} & \gamma\frac{\epsilon}{2} \\ 0 & (1 - \frac{\epsilon}{2})(\gamma - 1) \end{bmatrix} \begin{bmatrix} Q(\bar{a_1}) \\ Q(a_2^*) \end{bmatrix} + \begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}
$$

If we consider the right side of the above equations to be of the form $Mx + b$, then the solution to the following equation will find a fixed point:

$$
M_1 \begin{bmatrix} Q_{a_1} \\ Q_{a_2} \end{bmatrix} + \begin{bmatrix} R_{11} \\ R_{21} \end{bmatrix} = -c\left(M_2 \begin{bmatrix} Q_{a_1} \\ Q_{a_2} \end{bmatrix} + \begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix} \right).
$$

The left side is the equation for greedy action a_1^*, while the right is for greedy action a_2^*. The point will be fixed where the two vectors balance each other, scaled by a constant factor because one of the updates occurs more frequently. That is, any mixed strategy within the bounds of exploration is feasible on this line, but only one will keep the Q-values equal. In the tethered case, all Q-values are identical because the desired point is on the $Q_{a_1} = Q_{a_2}$ diagonal:

$$
\begin{aligned}
\mu &= (1 - \frac{\epsilon}{2})(1 - \gamma) \\
-\mu Q_{a_1} + R_{11} &= -c((\gamma - 1)\frac{\epsilon}{2}Q_{a_1} + R_{12}) \\
c &= \frac{\mu Q_{a_1} - R_{11}}{(\gamma - 1)\frac{\epsilon}{2}Q_{a_1} + R_{12}} \\
(\gamma - 1)\frac{\epsilon}{2}Q_{a_2} + R_{21} &= -c(-\mu Q_{a_2} + R_{22}).
\end{aligned}
$$

[1]Thank you to Satinder Singh for help with these insights.

Substituting for c, and setting $Q_{a_2} = Q_{a_1}$:

$$
(\gamma - 1)\frac{\epsilon}{2}Q_{a_1} + R_{21} = -\frac{\mu Q_{a_1} - R_{11}}{(\gamma - 1)\frac{\epsilon}{2}Q_{a_1} + R_{12}}(-\mu Q_{a_1} + R_{22})
$$

$$
\begin{aligned}
A &= \mu^2 - (\gamma - 1)^2 \frac{\epsilon^2}{4} = (1 - \gamma)^2(1 - \epsilon) \\
B &= -(\gamma - 1)\frac{\epsilon}{2}(R_{12} + R_{21}) - \mu(R_{11} + R_{22}) \\
C &= R_{11}R_{22} - R_{12}R_{21} \\
Q_{a_1} &= \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \\
Q_{a_1} &= \frac{-\frac{\epsilon}{2}(R_{12} + R_{21}) + (1 - \frac{\epsilon}{2})(R_{11} + R_{22})}{2(1 - \gamma)(1 - \epsilon)} \pm \frac{\sqrt{\frac{B^2 - 4AC}{(1 - \gamma)^2}}}{2(1 - \gamma)(1 - \epsilon)} \\
\frac{B^2}{(1 - \gamma)^2} &= ((1 - \frac{\epsilon}{2})(R_{11} + R_{12}) - \frac{\epsilon}{2}(R_{21} + R_{22}))^2 \\
\frac{4AC}{(1 - \gamma)^2} &= 4(1 - \epsilon)(R_{11}R_{22} - R_{12}R_{21}).
\end{aligned}
$$

This equation gives two fixed points. In both, the forces on either side of the $Q_{a_1} = Q_{a_2}$ line are equal and opposite. However, the fixed points on this line may not be stable. We know that a point is stable if the updates surrounding it push towards the point. Since the $Q_{a_1} = Q_{a_2}$ line has a slope of 1, the direction of a line crossing through the point will depend on whether the new line has a slope greater than 1. In general, the following theorem applies.

THEOREM 1. *Test of Stable Points: At the fixed points given above, the slope of the update below the line $Q_{a_1} = Q_{a_2}$, where $Q_{a_1^*} > Q_{\bar{a_2}}$, is defined by $k = \frac{\partial Q_{a_1^*}}{\partial t} / \frac{\partial Q_{\bar{a_2}}}{\partial t}$. If both or neither of the following conditions hold, then the examined point is stable. Otherwise, if only one holds, the point is unstable:*
(1) $k > 1$
(2) $\frac{\partial Q_{a_1^*}}{\partial t} > 1$.
An analogous test exists for updates above the line.

3.1 Finding the Boundary Between Regions of Attraction

Not all stable points are found on the $Q_{a_1} = Q_{a_2}$ line. If the game we're considering has a pure Nash equilibrium, the Q-values for the actions that constitute the Nash will represent a stable fixed point. All points in the space of Q-values will be attracted by some stable point, and we can bound the different regions in space where points are attracted to each stable point. Consider the case where there is one on-diagonal and one off-diagonal stable point, like in the Prisoner's Dilemma. One of them is on the $Q_{a_1} = Q_{a_2}$ line. The gradient at this point will necessarily be tangent to the $Q_{a_1} = Q_{a_2}$ line. If we project away from this point for $t < 0$ to find the contour leading into it, the projection defines the boundary between two basins of attraction:

$$
\begin{aligned}
M_2 \begin{bmatrix} Q_{a_1} \\ Q_{a_1} \end{bmatrix} + \begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix} &= \begin{bmatrix} k \\ k \end{bmatrix} \\
-(1 - \gamma)(1 - \frac{\epsilon}{2})Q_{a_1} + R_{22} &= (\gamma - 1)\frac{\epsilon}{2}Q_{a_1} + R_{12} \\
Q_{a_1} &= \frac{R_{22} - R_{12}}{(1 - \gamma)(1 - \epsilon)}.
\end{aligned}
$$

The equation that defines the dynamics for both Q-values above the $Q_{a_1} = Q_{a_2}$ line will show how the values behave
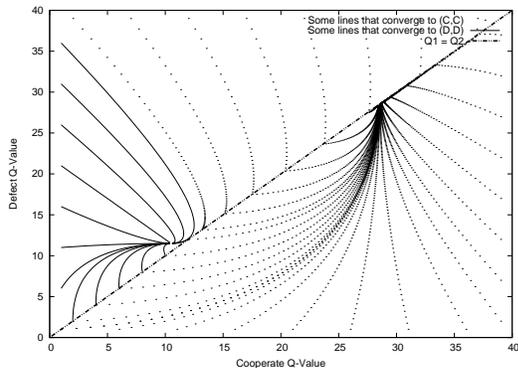
Figure 2: Update trajectories for IPD, given current Q-values for both actions. There are two regions corresponding to two attractor points. The update rule is IQL, $\gamma = 0.9$, and $\epsilon = 0.1$.

when $Q_{a_2} > Q_{a_1}$. Q_{a_2} as a function of t is given by the following differential equation, when $Q_{a_2} = Q_{a_2}(0)$ at $t = 0$:

$$R_{a_1^*} = (1 - \frac{\epsilon}{2})R(a_1^*, b_1^*) + \frac{\epsilon}{2}R(a_1^*, \bar{b_2})$$

$$R_{a_2^*} = (1 - \frac{\epsilon}{2})R(a_2^*, b_2^*) + \frac{\epsilon}{2}R(a_2^*, \bar{b_1})$$

$$\frac{\partial Q_{a_2}}{\partial t} = \delta_{a_2} = (1 - \frac{\epsilon}{2})((\gamma - 1)Q_{a_2}(t) + R_{a_2^*})$$

$$Q_{a_2}(t) = (Q_{a_2,0} - \frac{R_{a_2^*}}{1 - \gamma})e^{-\mu t} + \frac{R_{a_1^*}}{1 - \gamma}.$$

Using this information, we can compute Q_{a_1}, with the same methods. A quick check will confirm that substituting corresponding values and taking the derivative of the Q values will give the same answer:

$$\frac{\partial Q_{a_1}}{\partial t} = \delta_{a_1} = \frac{\epsilon}{2}(\gamma Q_{a_2} - Q_{a_1} + R_{a_1^*})$$

$$q = \frac{\frac{\epsilon}{2}\gamma(Q_{a_2,0} - \frac{R_{a_2^*}}{1-\gamma})}{\frac{\epsilon}{2} - \mu}$$

$$\rho = Q_{a_1,0} - q - \gamma\frac{R_{a_2^*}}{1 - \gamma} - R_{a_1^*}$$

$$Q_{a_1}(t) = \rho e^{-\frac{\epsilon}{2}t} + qe^{-\mu t} + \gamma\frac{R_{a_2^*}}{1 - \gamma} + R_{a_1^*}.$$

These equations and the point from above will provide the trajectory passing through the fixed point.

4. CONCLUSIONS AND FUTURE WORK

In this work, we showed how to find the fixed points where Q-values for actions were equal in the tethered multiagent case, as well as one way to find the regions attracted to these points. Q-learning with ϵ-greedy exploration is clearly different from Boltzmann exploration, as the actions shift discontinuously in the former case. We would like to provide a precise characterization of the behavior of IQL when the values are not tethered across agents. Does it converge for some initial Q-values, or is it always chaotic? Can we characterize the range of values encountered during the chaotic oscillations? We are interested in applying tools from non-linear dynamics to modify IQL to make it behave consistently. An appropriate tool could be inserted into Q-learning to modify

the exploration or learning rate to produce a more robust algorithm. It is apparent that the seeds of a powerful approach already exist in the simple form of Q-learning and we would like to provide a more reliable alternative.

5. REFERENCES

[1] R. Axelrod. *The Evolution of Cooperation*. Basic Books, 1984.

[2] M. Babes, E. Munoz de Cote, and M. L. Littman. Social reward shaping in the prisoner's dilemma. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1389–1392, 2008.

[3] M. H. Bowling. Convergence and no-regret in multiagent learning. In *NIPS*, 2004.

[4] J. Hu and M. P. Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069, 2003.

[5] M. L. Littman. Friend-or-foe Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 322–328. Morgan Kaufmann, 2001.

[6] M. L. Littman and P. Stone. Implicit negotiation in repeated games. In *Eighth International Workshop on Agent Theories, Architectures, and Languages (ATAL-2001)*, pages 393–404, 2001.

[7] E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 880–887, 2004.

[8] T. W. Sandholm and R. H. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:144–166, 1995.

[9] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 39:287–308, 2000.

[10] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of Uncertainty in AI (UAI)*, 2000.

[11] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[12] J. N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16(3):185–202, 1994.

[13] K. Tuyls, K. Verbeeck, and S. Maes. On a dynamical analysis of reinforcement learning in games: Emergence of occam's razor. *In Lecture Notes in Artificial Intelligence, Multi-Agent Systems and Applications III, LNAI 2691, CEEMAS 2003*, 2003.

[14] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.

[15] M. Zinkevich, A. R. Greenwald, and M. L. Littman. Cyclic equilibria in Markov games. In *Advances in Neural Information Processing Systems 18*, 2005.