

Multi Agent Systems, Lecture 5

Summary by Ola Rozenfeld

December 6, 2006

1 The Development Of Social Conventions (contd.)

Reminder: this section is based on the paper "[On The Emergence Of Social Conventions: Modeling, Analysis And Simulations](#)" by Yoav Shoham and Moshe Tennenholtz.

In the previous lecture we proved that the HCR algorithm in n -2-g social agreement games is guaranteed to eventually converge to a social convention. The focus of this section is to explore the efficiency with which such behavior is attained. First we present a formal definition of our efficiency measurement:

Definition 1 *Let g be a social agreement game. Consider an iteration t of a corresponding n -2-g iterative game, and the $n(n - 1)$ games (possible agent interactions) that might be played at that iteration. Let $X_n(t)$ be a random variable that contains the number of games that might be played at iteration t and which result in a payoff for a player that is less than the one obtained by a social convention. Let $T(n)$ be a function that associates with each n a number of iterations. Given a selection rule R and some distribution on the initial actions of the agents we will say that R guarantees the emergence of a rational social convention after $T(n)$ iterations, if $\lim_{n \rightarrow \infty} E[X_n(T(n))] = 0$.*

Remark: It is important to examine the above definition carefully in order to understand what exactly it means for fixed values of n . For example, if we have a rule R that, by the above definition, guarantees the emergence of a rational social convention after n^2 iterations, what does this fact imply on a 5-2-g iterative game? Is it guaranteed to converge to a social convention after 25 steps? No – since, in particular, there is a fixed probability that the same two agents will keep meeting in all first 25 iterations. Does it mean that $X_5(25)$ is bounded from above by some small value? No! So what does it mean? The answer is, that for any fixed n , the above concept means *nothing at all*. All it says is the following: *when n approaches infinity*, the expected amount of "bad" encounters will approach 0. It doesn't even mean, for example, that for $n = 6$ this number will be smaller than for $n = 5$, i.e. that $X_6(36) < X_5(25)$. All that the measure gives us is an asymptotic bound.

We now prove a lower bound on the value of $T(n)$:

Theorem 1 *Let g be a social agreement game and let R be an action selection rule. Assume that any action is played with some constant non-zero probability in the initial phase. If R guarantees the emergence of a rational social convention in the corresponding n-2-g game in $T(n)$ iterations, then $T(n) = \Omega(n \log(n))$.*

Proof: The idea behind the proof is that $T(n) = \Omega(n \log(n))$ is the minimal value that is required in order to ensure that all n agents participated in at least one round of the game until iteration $T(n)$ (by "ensure" we mean that the probability of an agent not being selected to play in $T(n)$ iterations approaches 0 as n grows). Formally, let $Y_n(i)$ be a random variable which contains the number of agents that did not participate in any iteration of n-2-g until iteration i . It is easy to see that $E[X_n(i)] \geq k \cdot E[Y_n(i)]$ for some constant $k > 0$ and for every n and i . In particular, $E[X_n(T(n))] \geq k \cdot E[Y_n(T(n))]$ for every n . Hence, it suffices to show that if $\lim_{n \rightarrow \infty} E[Y_n(T(n))] = 0$, then $T(n) = \Omega(n \log(n))$ (verify by contradiction). We will write $T(n)$ as $T(n) = (n-1) \cdot f(n)$. The probability that a particular agent will not be chosen along $T(n)$ iterations is bounded from below by $\left(1 - \frac{1}{n-1}\right)^{2 \cdot (n-1) \cdot f(n)}$ (observe that the probability not to choose a given agent in a single iteration is $\left(1 - \frac{1}{n}\right) \left(1 - \frac{1}{n-1}\right) \leq \left(1 - \frac{1}{n-1}\right)^2$). A classic calculus formula states that $\lim_{x \rightarrow \infty} \left(1 - \frac{1}{x}\right)^x = e^{-1}$; therefore, the probability of a particular agent not to be chosen in $T(n)$ iterations converges to $e^{-2f(n)}$. If $e^{-2f(n)} > \frac{1}{n}$ then we will get that $E[Y_n(T(n))] > 1$ for any n (since $E[Y_n(T(n))]$ is a sum of such probabilities over all agents) and hence there is no convergence to 0. But, in order to have $e^{-2f(n)} \leq \frac{1}{n}$ we must have $f(n) \geq 0.5 \cdot \log(n)$. This gives us the desired lower bound. ■

We can also show (proof not given in lecture) that the HCR learning rule guarantees the emergence of a rational social convention in an n-2-g game in $O(n \log(n))$ iterations, which makes it an optimal learning rule in the sense captured by the above study.

2 Optimal Teaching Policies

This section is based on the paper "[On Partially Controlled Multi-Agent Systems](#)" by Ronen Brafman and Moshe Tennenholtz.

Design of distributed algorithms assumes that the designer (programmer) has total control over all the agents in the system. Game theory, on the other hand, assumes that all agents behave rationally and independently. In this section we explore the situation of *partial control*: the situation where some agents are guaranteed to behave as prescribed, while others will behave according to their own best interests.

The first section of the paper (not presented in class) shows that the presence of controlled agents can force the entire agent population to adhere to a prescribed social convention, when the designer programs the controlled agents to include punishment mechanisms.

The section that we present here deals with the situation of a "teacher" (controlled agent) and a "student" (independent rational agent) that play an infinitely repeated game, in which the utility of the teacher depends only on the actions of the student. Therefore, the teacher would prefer the student to play a specific action – i.e. the goal of the teacher will be to "teach" the student to play the desired action. We assume that the game matrix is known to the teacher, but not to the student – therefore, the student will have to use some learning algorithm to discover the game.

In this section we will explore how to compute the optimal policy of the teacher given the learning algorithm used by the student. We will make some simplifying assumptions:

1. The game played is a $2 * 2 * g$ iterative game, where the payoffs to the student are given by:

	1	2
I	a	b
II	c	d

We denote the payoffs of the teacher by $u(I)$ and $u(II)$, and assume w.l.o.g. that $u(I) > u(II)$, i.e. the teacher is trying to teach the student to play I.

We can observe that the cases where either $a > c$ or $b > d$ are not interesting, since then the optimal teaching algorithm would be simply play 1 (resp. 2) constantly, which would make the student always observe a better payoff for I than for II. Therefore, the interesting setting occurs when $c > a$ and $d > b$. The case where $c > d > a > b$ is hopeless for the teacher, since no matter what complex policy he will choose, the student will always observe a better payoff for II than for I, and therefore he will never learn. So, the only interesting case (up to a permutation of the teacher's actions) is when $c > a > d > b$, which is exactly the Prisoner's Dilemma game.

2. The algorithm of the student is a finite state machine (we denote the state set by Σ), where each state $s \in \Sigma$ induces a probability ρ_s on the student's action space. Note the restrictions that this setting imposes: not only we assume that the student has a constant memory bound; the next state of the student is uniquely determined by his current state and the joint action of the players, while the action of the student is determined (albeit stochastically) only by his state.
3. The state machine of the student, as well as ρ , as well as the initial state, are all known to the teacher. Note that this implies that the teacher can always compute the student's current state.
4. The teacher's goal is to maximize his total discounted payoff, defined as follows: Let π be the teacher's policy, and assume that it induces a probability distribution $Pr_{\pi,k}$ over the set of possible student actions at time k . Then, the value of the policy π is:

$$val(\pi) = \sum_{k=0}^{\infty} \gamma^k E_k(u)$$

where

$$E_k(u) = \sum_{a \in A_s} Pr_{\pi,k}(a)u(a)$$

Here, $0 < \gamma < 1$ is some constant discount factor. This definition of payoff is commonly used to model situations in which the number of games played is unknown in advance; there, γ represents the probability that the game ends in each stage. Another "justification" of such approach is the decreasing value of goods – we might prefer to get \$20 today than \$21 tomorrow.

Given the above assumptions, we can show how to compute the optimal teaching policy. Some definitions are required:

Definition 2 A Markov Decision Process (MDP) is a tuple (S, A, P, r) , where

- S is a finite set of states
- A is a finite set of actions
- P is a probabilistic state transition function: $P : S \times S \times A \rightarrow [0, 1]$, where $P(s_1, s_2, a)$ is the probability of the transition from state s_1 to state s_2 given action a .
- r is the reward function: $r : S \rightarrow \mathfrak{R}$, where $r(s)$ is the reward from being in state s

Given an initial state $s \in S$ and some action policy $\pi : S \rightarrow A$, the distribution function $Pr_{s,\pi,k} : S \rightarrow [0, 1]$ can be defined, where $Pr_{s,\pi,k}(s')$ is the probability of getting to state s' after k steps, if we start from the initial state s and use policy π .

An action policy (strategy) is called γ_0 -optimal for a specific MDP if it maximizes the expected discounted reward $\sum_{k=0}^{\infty} \gamma_0^k (\sum_{s' \in S} Pr_{s,\pi,k}(s')r(s'))$. A classic theorem shows that such policy exists and can be efficiently computed (note that the policy does not depend on the initial state s).

In the next lecture we will show how finding an optimal teaching policy can be reduced to solving an appropriate MDP.