The Limits of Reinforcement Learning in Lewis Signaling Games

David Catteeuw
Artificial Intelligence Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels, Belgium
dcatteeu@vub.ac.be

Bernard Manderick
Artificial Intelligence Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels, Belgium
bmanderi@vub.ac.be

ABSTRACT

We study how communication systems can emerge through repeated interaction between two individuals. We apply three reinforcement learning algorithms (Roth-Erev learning, Learning Automata, and Q-Learning) to the repeated Lewis signaling game, a game theoretic model of a communication problem. Our experiments show that each of these algorithms always reach a state of optimal communication even for games with more than two types ($N > 2$) and for arbitrary type distributions.

We show how these algorithms are related and can be reduced to a simple random process which only updates behavior on success and never deviates from what was once successful. We call this random process "*win-stay/lose-inaction*".

For Lewis signaling games where the distribution over the types is uniform, the expected number of interactions needed by win-stay/lose-inaction to reach a state of optimal communication is a function of N: $1.45N^{3.02}$, where N is the size of the game –the number of types, signals, and responses.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Experimentation

Keywords

reinforcement learning, signaling, game theory, Lewis signaling games

1. INTRODUCTION

Communication or signaling is an important aspect of our everyday life. It is also an essential part of the interactions among and within all social organisms [10]. Examples of communication, or signaling, in nature include the alarm calls of Vervet monkeys [18], cooperative hunting of fish [6], the honey bees' waggle dance [22], and signaling among bacteria [11]. Signaling has played a major role in the self-organization of biological systems [14], and it is therefore desirable to have a good understanding of the mechanisms by which efficient signaling systems may be established.

The emergence of communication is also interesting from the viewpoint of artificial intelligence. The ability to signal is indeed advantageous for the coordination and cooperation between artificial agents. Even more so is the ability to bootstrap or adapt a communication system to the specific needs of the agent society. An example of a multi-agent system where signals acquire their meaning ex-nihilo, adapted to the specific needs of the problem at hand is given in [17].

In this paper we gain more insights into the capability of reinforcement learning to bootstrap communication. The reinforcement learning rules used in this text are discussed in Section 3. We study how arbitrary signals can acquire their meaning by applying reinforcement learning to the repeated Lewis signaling game. Philosopher David Lewis introduced signaling games in order to provide a theoretic approach to the problem of the emergence of conventions [13]. A convention, like a language, is a system of arbitrary rules that is shared between different agents, and that enables them to transfer information. A Lewis signaling game is a two-player game of common interest where the players are successful once they reach an agreement on the meaning of signals. We give more details on Lewis signaling games in Section 2.

The Lewis signaling game is an especially hard coordination game due to the existence of many suboptimal equilibria, so-called "pooling" or "partial pooling equilibria". Some algorithms, such as Roth and Erev's basic model [16], easily get stuck in these suboptimal equilibria [2, 12, 8]. Other learning rules, such as win-stay/lose-randomize, always lead to an optimal equilibrium in theory [3], but in practice require too much time for all but the smallest games [4, 3, 8]. We discuss the related work in Section 5.

A more acceptable solution to the problem of pooling equilibria should not introduce a random, but a guided drift towards more optimal signaling. Our main contribution is to experimentally show that Roth-Erev learning, Learning Automata, and Q-learning are able to overcome the problem of suboptimal equilibria and quickly find a state of optimal signaling precisely because they follow a guided drift, see Section 4. We also explain why this is the case. In extreme cases, these algorithms follow a heuristic we call "win-stay/lose-inaction" (Section 4.1). Just as win-stay/lose-randomize, win-stay/lose-inaction repeats behavior that is successful, but, whenever it is unsuccessful win-stay/lose-inaction does not alter its behavior, whereas win-stay/lose-randomize would choose an action at random next time.

A second contribution of this paper is the following. By analyzing the random process resulting from win-stay/lose-inaction we can predict the number of interactions needed to find a state of optimal communication. For Lewis signaling games with uniform type distributions the expected

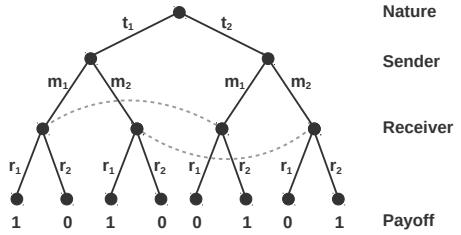
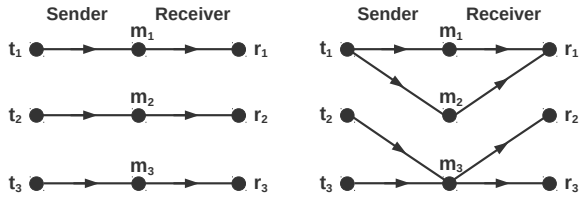


Figure 1: A Lewis signaling game with two states, signals and responses, $N = 2$. The game is played as explained in the text. A dashed line indicates that the decision nodes it connects cannot be distinguish from each other.



(a) An optimal equilibrium or signaling system

(b) A partial pooling equilibrium

Figure 2: Example agent configurations for a signaling game with 3 world states, signals and actions.

number of iterations needed to find a signaling system is $1.45N^{3.02}$, where N is the size of the game: the number of types, signals, and responses. Experiments with reinforcement learning confirm this result (Section 4.2). We discuss the importance of this result in Section 6.

Next, we discuss the Lewis signaling game and the notion of (partial) pooling equilibria in greater detail.

2. THE LEWIS SIGNALING GAME

A Lewis signaling game [13] is a two-player extensive form game such as the one shown in Figure 1. The first player (called *Sender*) has some information (its *type*) which the second player (called *Receiver*) cannot directly observe. However, Sender can send a signal which Receiver can observe. It is in both players' interest that Receiver can reliably deduce Sender's type.

More formally, the game is played as follows. First, Nature determines the type t_i of Sender, by randomly drawing from a discrete probability distribution π. The type is private information of Sender. Next, based on his type, Sender sends a signal m_j to Receiver. Receiver, in turn, selects a response r_k. The number of types, signals, and responses is N. So i, j, and $k \in \{1, 2, \ldots, N\}$. Finally, Sender and Receiver get a payoff u depending on Sender's type t_i, and Receiver's response r_k as follows:

$$u = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

So, for each type t_i, there is exactly one correct response r_i, and each response r_k is optimal for exactly one type t_k. The payoff u does not depend on the signal sent. Since there is an equal number of types, signals, and responses, a Lewis signaling game is completely determined by its type distribution π. Note that, although we index types, signals, and responses with numbers, the agents cannot rely on this order. Instead of using Equation (1), we could as well use any other one-to-one correspondence between types and responses, and this would not change the game. This is a standard requirement in game theory.

The essence of the game is that agents face a communication problem, which they can solve by establishing a shared language or convention –that is to say, by adopting compatible mappings from types to signals (by Sender) and from signals to responses (by Receiver). Their mappings will be compatible if they, when applied one after the other (first Sender's, then Receiver's mapping), lead to the correct response for all types. For a Lewis signaling game with N types, signals, and responses there are $N!$ different but equally valid and optimal conventions possible, corresponding to the $N!$ unique mappings from types to signals, or from signals to responses. One such convention is shown in Figure 2a for $N = 3$. In that case, the *signaling success rate*, which we define as the probability that the two agents will have a successful interaction, is 1. Since the payoff for success is 1, and the payoff for failure is 0, the expected payoff for following a shared convention also is 1. Lewis calls such an optimal equilibrium a *signaling system*.

A Lewis signaling game also has many other equilibria, which are suboptimal, especially when the number of types is larger than two, $N > 2$. Such equilibria are called "pooling" or "partial pooling equilibria", because they involve Sender using the same signal for different types. Figure 2b shows a partial pooling equilibrium where Sender uses signal m_3 both for type t_2 and t_3. As a consequence, Receiver, when observing signal m_3 can only guess what is the true type of Sender. Assuming that all types are equally likely, the signaling success rate (and the expected payoff) for the agents is 2/3: for type t_1 there is always success, for type t_2 and t_3 there is only success half of the time. This state is an equilibrium, since, neither Sender nor Receiver can change his mapping to increase his payoff. Sender's mapping is a best response to Receiver's mapping, and vice versa, Receiver's mapping is a best response to Sender's mapping. It is the existence of many such suboptimal equilibria that makes it hard to find an optimal one.

Another difficulty arises when the type distribution π is non-uniform. For example, if one of the types has a probability of 90%, Receiver can simply ignore the signals, always pick the action corresponding to the most frequent state, and the signaling success rate will already be as high as 90%.

As mentioned in Section 1, some learning algorithms may get stuck in these suboptimal equilibria and never reach an optimal one. Others are guaranteed to find an optimal equilibrium in theory, but require too much time in practice. See more related work in Section 5.

In [8], the authors showed experimentally that Roth-Erev learning with forgetting is able to overcome both difficulties and find a convention reasonably fast. Here we show, that Q-learning, and Learning Automata can have similar performances, we explain why, and give a formula for the expected number iterations needed to reach a convention in Lewis signaling games with uniform type distributions. Next, we explain these reinforcement learning rules and how we apply them to Lewis signaling games.

3. REINFORCEMENT LEARNING

In this text we consider three simple reinforcement learning rules: Roth-Erev learning [16], Q-learning [23], and Learning Automata [15]. Roth-Erev learning and Q-learning are so-called action-value methods. For each state-action pair (s, a) these algorithms have a value $q_{s,a}$ which indicates the quality of taking action a in state s relative to the other actions. These methods consist of an update rule and an action selection rule. The update rule determines how action values are updated based on new experience. The action selection rule determines which action to select, given the current state and the action values, by calculating the probability $p_{s,a}$ of taking action a in the current state s for all actions a. The usual constraints on probabilities hold: $\forall s : \sum_a p_{s,a} = 1$ and $\forall s, a : p_{s,a} \geq 0$. The basic idea is that action values of successful actions increase and actions with higher values get selected more often than actions with lower action values. Learning Automata, on the other hand, directly update a probability distribution over the actions, and select actions according to this distribution.

To apply these algorithms to signaling games, Sender will have an action value $q_{t,m}$ and a probability $p_{t,m}$, for all types t and signals m. Likewise, Receiver will have an action value $q_{m,r}$ and a probability $p_{m,r}$, for all signals m and responses r. So, signals take the role of actions for Sender, and the role of states for Receiver. Of course, Learning Automata only have probabilities $p_{s,a}$ for all state-action pairs (s, a) and no action values $q_{s,a}$.

We shortly give the details of the reinforcement learning algorithms. The first algorithm, *Roth-Erev learning* [16], work as follows. At each time step, given the current state s, the probability of selecting action a ($p_{s,a}$) is proportional to the action value $q_{s,a}$:

$$p_{s,a} = \frac{q_{s,a}}{\sum_{a'} q_{s,a'}} \tag{2}$$

This assumes that all action values (and consequently, payoffs) are positive, $\forall s, a : q_{s,a} \geq 0$. If all action values are 0, each action is selected with equal probability. After taking action a and receiving payoff u, all action values for the current state s are discounted by factor λ and only the action value of the current action is incremented with payoff u:

$$q_{s,a} \leftarrow \begin{cases} \lambda q_{s,a} + u & \text{if action } a \text{ was taken,} \\ \lambda q_{s,a} & \text{otherwise.} \end{cases} \tag{3}$$

Action values for states other than the current one are not updated. This algorithm has two parameters: the discount factor $\lambda \in [0, 1]$ and the initial action-value $q(0) \geq 0$. In Roth and Erev's basic model both are set to 1, $\lambda = q(0) = 1$. The idea of setting the discount factor λ to a value less than 1, is that it helps forgetting old experience. Mathematically, it bounds the action-values to $u/(1 - \lambda)$, which makes it possible for the algorithm to settle down. If the discount factor $\lambda = 1$ the action values are unbounded and the system never settles down. Using small initial action values, $q(0) < u$, speeds up learning in the beginning because it increases the importance of the payoffs u.

The second algorithm we use is *Q-learning* [23] which only updates the action value of the action that was actually chosen in the current state s:

$$q_{s,a} \leftarrow \begin{cases} q_{s,a} + \alpha(u - q_{s,a}) & \text{if action } a \text{ was taken,} \\ q_{s,a} & \text{otherwise,} \end{cases} \tag{4}$$

where $\alpha \in [0, 1]$ is the learning rate. A higher learning rate puts more weight on more recent payoffs. If α is 0, nothing is ever learned, if α is 1, the action value of an action simply equals the last payoff earned for that action. A very popular action-selection strategy for Q-learning is ϵ-greedy action selection. It selects an action at random, with probability ϵ, and selects an action with the highest action value with probability $1 - \epsilon$. If $\epsilon = 0$ then there is no exploration, and we simply call this "greedy action-selection". This algorithm has three parameters: the learning rate $\alpha \in [0, 1]$, the exploration rate $\epsilon \in [0, 1]$, and the initial action value $q(0) \in \mathbb{R}$. Higher initial action values will increase the amount of exploration in the beginning [21, p39].

Finally, *Learning Automata* [15] directly update the probability distribution over the actions. There are different well-known schemes, one of which is Linear-Reward-Inaction (L_{R-I}). This learning algorithm only updates the action probabilities on reward, that is when the payoff $u > 0$:

$$p_{s,a} \leftarrow \begin{cases} p_{s,a} + \alpha u(1 - p_{s,a}) & \text{if action } a \text{ was taken,} \\ p_{s,a} - \alpha u p_{s,a} & \text{otherwise,} \end{cases} \tag{5}$$

where $\alpha \in [0, 1]$ is again the learning rate. The learning rule requires that the payoff u belongs to the interval $[0, 1]$, which is the case for Lewis signaling games. The initial probability distribution over the actions is uniform for all states s, so the learning rate $\alpha \in [0, 1]$ is the only parameter of the algorithm.

4. RESULTS

We now turn to some experimental results with these algorithms. We concentrate on two performance criteria: (a) whether or not a convention is reached, and (b) the number of iterations needed to reach a convention. Figure 3 shows the results for the various algorithms and different parameters for the Lewis signaling game with type distribution $\pi = \langle \frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \frac{4}{36}, \frac{5}{36}, \frac{6}{36}, \frac{7}{36}, \frac{8}{36} \rangle$. Per experiment, we did 1,000 runs. Per run we recorded how many iterations were needed to reach a convention. If, after 100,000 iterations, still no convention was reached, we stopped the run and counted it as a failure.

We say that Sender and Receiver reach a convention if the signaling success rate is above some *threshold* θ. Remember that the signaling success rate, is the probability that the next game will be successful. We do not set the threshold to $\theta = 1$. This would be too strict, since some strategies can never reach this level although they perform almost optimal. ϵ-greedy Q-learning for example can never achieve a signaling success rate of 1, unless the exploration rate $\epsilon = 0$. Therefore, we choose to set the threshold θ halfway between the optimal value, which is 1, and the signaling success rate of the best suboptimal equilibrium. This allows to clearly distinguish between runs which do not end up in an optimal state but are very near and runs that are closer to a suboptimal equilibrium than an optimal one.

The *best suboptimal equilibrium* has a signaling success rate of 1 minus the probability of the least occurring type: $1 - \min_t(\pi_t)$, where π_t is the probability that Nature draws type t. The best suboptimal equilibrium is actually a partial pooling equilibrium such as the one in Figure 2b, where Sender uses the same signal m_3 for different types. When seeing this signal m_3, Receiver cannot distinguish between the types t_2 and t_3, and hence can do no better than as-

suming the most frequent type. So whenever *the other* type occurs the game will fail. In all other cases, the game succeeds. Thus, the signaling success rate in the *best* partial pooling equilibrium is determined by the frequency of the *least* frequent type. An example of an experiment is shown in Figure 3 for the Lewis signaling game with type distribution $\pi = \langle \frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \frac{4}{36}, \frac{5}{36}, \frac{6}{36}, \frac{7}{36}, \frac{8}{36} \rangle$. For this example, the least frequent type is $t_1 = min_t(\pi_t)$ with probability $\pi_{t_1} = 1/36$. So, the threshold is at $\theta = 71/72$, halfway between 1 and 35/36.

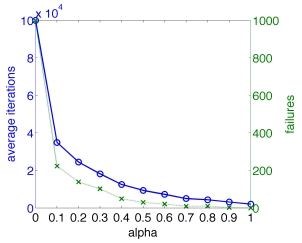
The experiments tell us the following:

- The learning automata perform better (both in terms of time needed and probability to fail) for higher learning rates, and best when the learning rate $\alpha = 1$.
- Q-learning performs best when playing greedy (exploration rate $\epsilon = 0$). Any learning rate $0 < \alpha < 1$ is fine.
- As we already showed in [8], Roth-Erev learning performs best with a small discount-factor λ, but a discount-factor of $\lambda = 0$ is bad. Roth-Erev learning with $\lambda = 0$ is the same as win-stay/lose-randomize, which, although it always reaches a convention in theory, is clearly too slow in this example since none of the 1,000 runs reached a convention in less than 100,000 iterations.
- Roth-Erev learning also performs better when initial action values are very small, and best when they are zero, $q(0) = 0$. This was also reported by Skyrms [19, p97].
- Finally, we notice that all three algorithms perform equally well in the best case. For the example in Figure 3, on average, for optimal parameters, these algorithms need slightly more than 2,000 iterations, and none of the 1,000 runs failed to find a convention in less than 100,000 iterations. For other Lewis signaling games we find similar results. As you may expect, the more types, the more time needed to find a convention and non-uniform type distributions are harder than uniform type distributions.

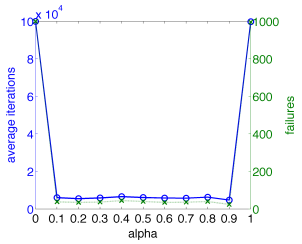
4.1 Win-stay/lose-inaction

We can explain the good performance of these algorithms by showing their similarity to a random process we call "win-stay/lose-inaction". Let us first introduce this simple process and explain how it behaves in the Lewis signaling game. Win-stay/lose-inaction is the following process:

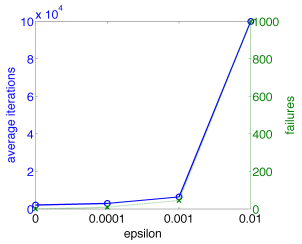
- The initial behavior in each state is to choose any action with equal probability, $p_{s,a} = 1/N$ for all state-action pairs (s, a).
- Whenever an interaction results in a failure (payoff $u = 0$), the behavior remains as it was (the action probabilities $p_{s,a}$ do not change).
- Whenever an interaction results in a success (payoff $u = 1$), the agents will remember what they did in the current state s. Whenever they end up in this state again, they use the action a that yielded success. So, probability $p_{s,a} = 1$, and the probabilities $p_{s,a'} = 0$ for all other actions $a' \neq a$.



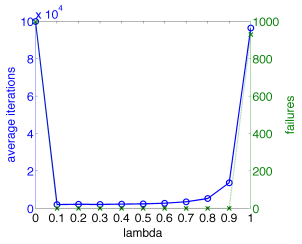
(a) Linear-Reward-Inaction learning automata for different learning rates α.



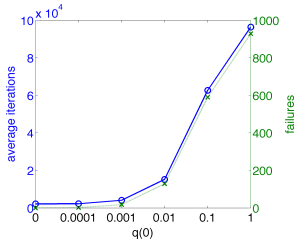
(b) ϵ-greedy Q-learning for different learning rates α and exploration rate $\epsilon = 0.001$.



(c) ϵ-greedy Q-learning for different exploration rates ϵ and learning rate $\alpha = 0.1$.



(d) Roth-Erev learning for different discount factors λ and initial action-value $q(0) = 1$.



(e) Roth-Erev learning for different initial action values $q(0)$ and discount factor $\lambda = 1$.

Figure 3: The four figures show the number of iterations needed to learn a convention averaged over 1,000 runs (blue, solid lines with circles, left y-axis), and the number of runs (out of 1,000) that failed to find a convention in less than 100,000 iterations (green, dashed line with crosses, right y-axis) for the Lewis signaling game with type distribution $\pi = \langle \frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \frac{4}{36}, \frac{5}{36}, \frac{6}{36}, \frac{7}{36}, \frac{8}{36} \rangle$.

We explain how win-stay/lose-inaction learns a convention in a Lewis signaling game with $N = 2$ types, signals, and responses. Figure 4a shows the initial behavior for both Sender (from types t to signals m) and Receiver (from signals m to responses r). Initial behavior is entirely random and the behavior will remain this way as long as there was no successful interaction. When at some point in time, Receiver happens to choose the correct response for the current type, then there will be a first success. Without loss of generality, we can label the current type as t_1, and the current signal as m_1. Remember that the ordering of types, signals, and responses is of no concern to the agents. Sender will, from now on, always use signal m_1 when observing type t_1 and Receiver will always respond with r_1 to signal m_1. We say a path $t_1 \to m_1 \to r_1$ is learned for type t_1. Behavior for other types and signals remains entirely random. This situation is shown in Figure 4b. Several things can happen now.

1. Nature draws type t_1. This will trigger Sender to send signal m_1, and consequently, Receiver will respond with r_1. So, this will always lead to success, and the path $t_1 \to m_1 \to r_1$ persists. In general, whenever a type occurs for which a path was already learned, the interaction is successful and the agents do not change their behavior.

2. Nature can also draw t_2, or in general, a type for which no path is yet learned. Sender will now pick a signal at random and there are again two possibilities:

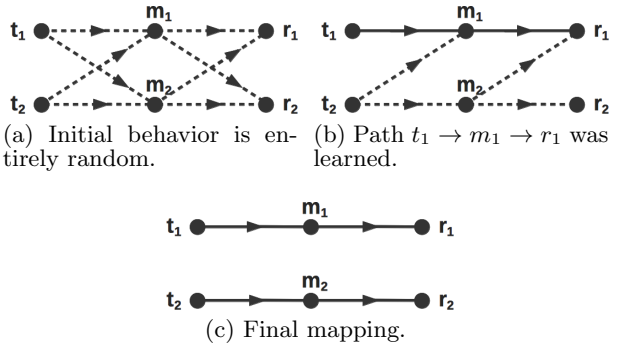
   (a) If he picks signal m_1 (or in general, a signal which is already used in a learned path), then Receiver will definitely respond with r_1 and the game fails. The agents do not update their behavior in that case.

   (b) If, on the other hand, Sender chooses m_2, (or in general, any signal which is not yet used in a path), then either Receiver guesses the incorrect response, which results in a failure, or he guesses the correct response. In the former case, the behavior of both agents remains the same. In the latter case, they will update their behavior and a new path $t_2 \to m_2 \to r_2$ will complete the convention.

We are now in the situation as shown in Figure 4c and from now on each interaction will be successful. It is straightforward to generalize the reasoning above to all Lewis signaling games.

When analyzing the three reinforcement learning rules from Section 3, we can see that they behave exactly as win-stay/lose-inaction. For the Learning Automata this is the case when the learning rate $\alpha = 1$. When an interaction fails, payoff $u = 0$, the probabilities over the actions don't change, see Equation (5). When there is success, payoff $u = 1$, the probability of the action taken, becomes 1, and all others 0, independent of the current probabilities. Also, initial behavior is random.

Roth-Erev learning equals win-stay/lose-inaction when the initial action values $q(0) = 0$, and discount factor $0 < \lambda \leq 1$. Q-learning reduces to win-stay/lose-inaction when the initial action values $q(0) = 0$, the learning rate $0 < \alpha < 1$, and the exploration rate $\epsilon = 0$. For Roth-Erev and Q-learning, it is somewhat harder to see this, because

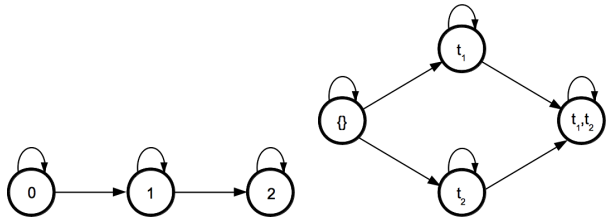


(a) Initial behavior is entirely random.

(b) Path $t_1 \to m_1 \to r_1$ was learned.

(c) Final mapping.

Figure 4: Evolution of learning by win-stay/lose-inaction for a Lewis signaling game with $N = 2$ types. A solid line represents a probability of 1. A dashed line represents a probability of $1/N$.



(a) Uniform type distributions.

(b) General.

Figure 5: The Markov chains for Lewis signaling games with $N = 2$ types.

the action values do *change on failure* (the action value of the action resulting in failure is slightly decreased), but it would take an infinite number of failures before the probability distribution over the actions also changes.

4.2 Number of interactions needed to reach convention

Now that we understand how win-stay/lose-inaction behaves in Lewis signaling games we can actually predict the number of iterations needed, on average, to reach a convention. We can identify the learning with a Markov chain where each state is uniquely represented by the set of types for which a path is already learned. Figure 5b shows the Markov chain for Lewis signaling games with two types ($N = 2$). There is one initial state where none of the types has a path. There is also one final state where all types have a path. Furthermore, there are N states where one type has a path, etc. Since, win-stay/lose-inaction can never forget a learned path, the Markov chain can never get into a state with fewer learned paths than the current state. For the same reason, the random process either remains in the same state, or it goes to a state where one extra path is learned. The probability to go to a next state is the probability that a new path is learned. It is the product of the probability of Nature drawing a type for which no path yet exists, the probability of Sender using a signal which is not yet used in a path, and the probability of Receiver picking the correct response. The latter is always $1/N$, where N is the number of responses.

For Lewis signaling games with uniform type distributions and N types, it does not matter for which type a path is

learned. The number of types for which a path is learned is sufficient to discriminate all states of the Markov chain, see Figure 5a. So, the probability p_l of learning a new type-signal-response path only depends on the number of paths already learned:

$$p_l = \frac{N-l}{N}\frac{N-l}{N}\frac{1}{N} = \frac{(N-l)^2}{N^3} \tag{6}$$

where l is the number of paths already learned. The first factor is the probability of Nature choosing a type t_i for which no path is learned, $(N-l)/N$, the second factor is the probability of Sender selecting a signal which is not yet used in a learned path, $(N-l)/N$, and the third factor is the probability that Receiver chooses the action a_i corresponding to the current type t_i, $1/N$. This probability p_l corresponds to the second part of case 2b of the analysis in Section 4.1. In all other cases, the Markov process remains in the same state.

The number of iterations needed before a new path is learned is distributed according to a geometric distribution which has mean $\mu = 1/p$ and variance $\sigma^2 = (1-p)/p^2$, where p is the probability to learn a new path. The expected number of iterations $E[T_c]$ to learn a complete mapping for a Lewis signaling game with N types and uniform type distribution π is the sum of the expected number of iterations needed for each new path:

$$E[T_c] = \sum_{l=0}^{N-1} \frac{1}{p_l} = \sum_{l=0}^{N-1} \frac{N^3}{(N-l)^2}. \tag{7}$$

We computed the expected number of iterations until optimal signaling is reached ($E[T_c]$) for different number of types (N) by means of Equation (7) and applied a standard linear regression analysis in log-log scale. In other words, the logarithm of the expected number of iterations required to reach a state of optimal signaling is a linear function of the logarithm of the size of the game: $\log E[T_c] = A + B \log N$. Converting back to linear scale, yielded the following result: $E[T_c] \approx 1.45N^{3.02}$. This agrees very well with the experimental data generated by applying the three reinforcement learning rules to Lewis signaling games with different number of types but uniform type distributions. Figure 6 shows the results for Linear-Reward-Inaction Learning Automata with learning rate $\alpha = 1$. Regression analysis on this data predicts that the number of iterations needed before reaching optimal signaling is $E[T_c] \approx 1.21N^{3.10}$.

In a similar way one can calculate the expected time $E[T_c]$ needed to reach a convention in Lewis signaling games with general type distributions π. The added difficulty is that the probabilities of going from one state to another now also depend on the order in which the paths for types are learned. This allows for multiple trajectories through the Markov chain from the initial state to the state where a complete mapping is learned, one per possible way of ordering the types. That is, $N!$ in total, assuming all types have a different probability.

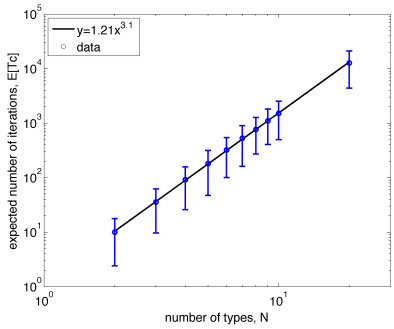


Figure 6: Experimental results of Learning Automata with learning rate $\alpha = 1.0$ showing the iterations needed to reach full coordination in Lewis signaling games with uniform type distributions. Each point shows the average number of iterations needed (over 10,000 simulations) and the standard deviation over the data. The confidence interval around the average is too small to show.

5. RELATED WORK

Reinforcement learning in Lewis signaling games.

Argiento et al. [1] have proven that basic *Roth-Erev learning* (with initial action values $q(0)$ and discount factor λ equal to one) always converges to a signaling system if the number of types $N = 2$ and if both types occur with the same frequency, $\pi = \langle \frac{1}{2}, \frac{1}{2} \rangle$. Basic Roth-Erev learning fails whenever the number of types $N > 2$ or when the probability distribution π over the types is not uniform [2, 12, 8]. Skyrms reported that smaller initial action values ($q(0) < 1$) increase the probability of converging to a signaling system [19, p97], even when the number of types is larger than two ($N > 2$), and the probability distribution over the types π is non-uniform. Here, we were able to give an explanation of why this is the case. Barrett and Zollman apply win-stay/lose-randomize to Lewis signaling games and prove that it always reaches a signaling system [3]. Win-stay/lose-randomize is equal to Roth-Erev learning when the discount factor $\lambda = 0$. Although this is theoretically very interesting, experiments show that the number of interactions needed to reach a signaling system increases exponentially with the size of the game [8]. Catteeuw et al. [8] show experimentally how Roth-Erev learning with a discount factor $0 < \lambda < 1$ always reaches a signaling system and is much faster than win-stay/lose-randomize [3]. Barrett [2] studied two other variations of Roth-Erev learning for signaling games with an arbitrary number of types ($N \geq 2$) but uniform type distributions. One variation allows for negative rewards, the other randomizes action values. Both variations seem to help reaching a signaling system, but do not guarantee it.

Barrett and Zollman [3] also discuss *Bush-Mosteller learning* [7] (which has the same update rule as Q-learning, Equation (4)) and a more complex learning rule called "*ARP*" [5]. For both Bush-Mosteller and the ARP model, the experiments show convergence to a signaling system is more certain if the learning parameters are correctly tuned.

Win-stay/lose-shift (called "best response" in [19]) does not work in Lewis signaling games with $N = 2$ types, since the process may get into endless loops. When there are more than two types ($N > 2$), the process must be redefined. For example, when loosing, we could pick any of the alternatives at random with equal probability. Skyrms calls this process *best response for all we know* [19, pp103-105]. Still, this process cannot handle the case for $N = 2$ types, and Zollman proposes to add inertia: only now and then apply the best response rule. The rest of the time behavior is not updated.

Steels' language games.

The emergence of a shared lexicon between multiple individuals has been extensively studied before in the domain of language evolution by means of language games, see for example [20].

The guessing game [9] is probably the language game which resembles the Lewis signaling game the most. It is played by two agents. The first is called Speaker, the second Hearer. At each interaction Speaker and Hearer are presented with a set of objects. This is called the *context* and varies from interaction to interaction. First, Speaker utters a word referring to one of the objects. Next, Hearer guesses which object Speaker refers to. Optionally, Speaker gives feedback to Hearer indicating whether or not he pointed to the correct object. In the guessing game, such feedback is indeed optional and is not at all necessary for learning. Since the context changes from interaction to interaction, Hearer may infer from that context what a word can and cannot mean. The notion of a context is typical for language games and is an important difference with signaling games. A second difference is the following. Although, in signaling games, there is a one-to-one mapping between types and responses, this is not clear to the agents. In language games, on the other hand, types and responses are both the same thing: namely objects. It is assumed that the agents know this and are allowed to exploit it.

6. CONCLUSION

In order to gain more insight into the emergence of communication and more particularly the capabilities of reinforcement learning to bootstrap a communication system, we studied three reinforcement learning algorithms in Lewis signaling games.

In summary, we found that these reinforcement learning algorithms always find a convention and do this reasonably fast, avoiding the many suboptimal equilibria even for Lewis signaling games with more than two types ($N > 2$) and non-uniform type distributions. The algorithms are Linear-Reward-Inaction Learning Automata with learning rate $\alpha = 1$, greedy Q-learning with initial action value $q(0) = 0$ and learning rate $0 < \alpha < 1$, and Roth-Erev learning with initial action value $q(0) = 0$ and learning rate $0 < \lambda \leq 1$. Second, we show that at best, these algorithms perform like win-stay/lose-inaction. Third, we analyzed win-stay/lose-inaction and gave a formula for the expected number of iterations before a convention is reached for Lewis signaling games with uniform type distributions: $E[T_c] = 1.45N^{3.02}$. Experiments confirmed this for the three reinforcement learning algorithms.

This result is important from two points of view. First, it tells us that under very weak assumptions communication can emerge and can do this reasonably fast. The number of interactions necessary to reach optimal communication is only polynomial in the number types, signals, and responses, N. Note that while learning there is already partial success. Of course, for really large communication systems, one would prefer time grows linear or even sublinear with the number of types N, and so other solutions are necessary. For example, if an agent could know which signal he uses for which type, he could avoid using that same signal for other types. This would definitely speed up learning, but it imposes extra requirements on the capabilities of the agents. This is exactly what researchers in the domain of language games are doing.

Second, this result is also interesting to the designer of a multi-agent system who has now an idea of what kind of learning rules work well and how many interactions his agents would be need to create an optimal signaling system.

You could argue that the learning entirely depends upon the rewards, and that win-stay/lose-inaction would stop working if the payoffs would change from $u = 1$ for success, and $u = 0$ for failure, to, for example, $u = 2$ and $u = 1$ respectively. According to game theory, however, payoffs only express the agents' preference order of different outcomes and the exact value is in fact internal to the agent. So, the agent's learning mechanism is free to manipulate a payoff before using it to update his action values. A simple way to do that is by a so-called reference point as used by Bush-Mosteller learning. If the payoff is below this reference point, the agent uses 0, if the payoff is above the reference point, the agent uses 1. This way we obtain again the original payoffs.

As future work, we would like to extend this work to populations of learning agents. You may expect that simply applying win-stay/lose-inaction in populations will not allow all agents to use the same "language". Preliminary experiments, however, show that a simple birth-death process where badly performing agents die, and new-born agents appear, may help to spread a single language through the entire population.

7. REFERENCES

[1] R. Argiento, R. Pemantle, B. Skyrms, and S. Volkov. Learning to signal: Analysis of a micro-level reinforcement model. *Stochastic Processes and their Applications*, 119(2):373–390, Feb. 2009.

[2] J. A. Barrett. Numerical Simulations of the Lewis Signaling Game: Learning Strategies, Pooling Equilibria, and the Evolution of Grammar. Technical Report September, University of California, Irvine: Institute for Mathematical Behavioral Science, 2006.

[3] J. A. Barrett and K. J. S. Zollman. The role of forgetting in the evolution and learning of language. *Journal of Experimental & Theoretical Artificial Intelligence*, 21(4):293–309, Dec. 2009.

[4] A. W. Beggs. On the convergence of reinforcement learning. *Journal of Economic Theory*, 122(1):1–36, May 2005.

[5] Y. Bereby-Meyer and I. Erev. On Learning To Become a Successful Loser: A Comparison of Alternative Abstractions of Learning Processes in the Loss Domain. *Journal of mathematical psychology*, 42(2/3):266–286, June 1998.

[6] R. Bshary, A. Hohner, K. Ait-el Djoudi, and H. Fricke. Interspecific communicative and coordinated hunting between groupers and giant moray eels in the Red Sea. *PLoS biology*, 4(12):e431, Dec. 2006.

[7] R. R. Bush and F. Mosteller. A mathematical model for simple learning. *Psychological review*, 58(5):313–23, Sept. 1951.

[8] D. Catteeuw, J. De Beule, and B. Manderick. Roth-Erev Learning in Signaling and Language Games. In P. De Causmaecker, J. Maervoet, T. Messelis, K. Verbeeck, and T. Vermeulen, editors, *Proceedings of the 23rd Benelux Conference on*

Artificial Intelligence, pages 65–74, Ghent, Belgium, 2011.

[9] J. De Beule, B. De Vylder, and T. Belpaeme. A cross-situational learning algorithm for damping homonymy in the guessing game. In L. M. Rocha, L. S. Yaeger, M. A. Bedeau, D. Floreano, R. L. Goldstone, and A. Vespignani, editors, *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 466–472. MIT Press, 2006.

[10] P. D'Ettorre and D. P. Hughes. *Sociobiology of communication*. Oxford University Press, New York, NY, 2008.

[11] G. M. Dunny and S. C. Winans. *Cell-cell signaling in bacteria*. American Society for Microbiology, 1999.

[12] S. M. Huttegger. Evolution and the Explanation of Meaning. *Philosophy of Science*, 74(1):1–27, Jan. 2007.

[13] D. K. Lewis. *Convention: A Philosophical Study*. Harvard University Press, Cambridge, 1969.

[14] J. Maynard Smith and E. Szathmary. *The major transitions in evolution*. Oxford University Press, 1997.

[15] K. S. Narendra and M. A. L. Thathachar. *Learning automata: an introduction*. Prentice-Hall, Upper Saddle River, NJ, USA, 1989.

[16] A. E. Roth and I. Erev. Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. *Games and Economic Behavior*, 8(1):164–212, 1995.

[17] A. Servin and D. Kudenko. Multi-Agent Reinforcement Learning for Intrusion Detection : A Case Study and Evaluation. In *6th German Conference on Multi-Agent System Technologies*, pages 159–170, 2008.

[18] R. M. Seyfarth, D. L. Cheney, and P. Marler. Vervet monkey alarm calls: Semantic communication in a free-ranging primate. *Animal Behaviour*, 28(4):1070–1094, Nov. 1980.

[19] B. Skyrms. *Signaling: Evolution, Learning and Information*. Oxford University Press, New York, 2010.

[20] L. Steels. Language Games for Autonomous Robots. *IEEE Intelligent Systems*, September:17–22, 2001.

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

[22] K. Von Frisch. *The dance language and orientation of bees*. Harvard University Press, 1967.

[23] C. J. C. H. Watkins and P. Dayan. Q-Learning. *Machine Learning*, 8:279–292, 1992.