7

Learning and Teaching

The capacity to learn is a key facet of intelligent behavior, and it is no surprise that much attention has been devoted to the subject in the various disciplines that study intelligence and rationality. We will concentrate on techniques drawn primarily from two such disciplines—artificial intelligence and game theory—although those in turn borrow from a variety of disciplines, including control theory, statistics, psychology and biology, to name a few. We start with an informal discussion of the various subtle aspects of learning in multiagent systems and then discuss representative theories in this area.

7.1 Why the subject of "learning" is complex

The subject matter of this chapter is fraught with subtleties, and so we begin with an informal discussion of the area. We address three issues—the interaction between learning and teaching, the settings in which learning takes place and what constitutes learning in those settings, and the yardsticks by which to measure this or that theory of learning in multiagent systems.

7.1.1 *The interaction between learning and teaching*

Most work in artificial intelligence concerns the learning performed by an individual agent. In that setting the goal is to design an agent that learns to function successfully in an environment that is unknown and potentially also changes as the agent is learning. A broad range of techniques have been developed, and learning rules have become quite sophisticated.

In a multiagent setting, however, an additional complication arises, since the environment contains (or perhaps consists entirely of) other agents. The problem is not only that the other agents' learning will change the environment for our protagonist agent—dynamic environments feature already in the single-agent case—but that these changes will depend in part on the actions of the protagonist agent. That is, the learning of the other agents will be impacted by the learning performed by our protagonist.

The simultaneous learning of the agents means that every learning rule leads to a dynamical system, and sometimes even very simple learning rules can lead to complex global behaviors of the system. Beyond this mathematical fact, however, lies a conceptual one. In the context of multiagent systems one cannot separate

	L	R
T	1, 0	3, 2
B	2, 1	4, 0

Figure 7.1 Stackelberg game: player 1 must teach player 2.

learning and teaching

the phenomenon of *learning* from that of *teaching*; when choosing a course of action, an agent must take into account not only what he has learned from other agents' past behavior, but also how he wishes to influence their future behavior.

The following example illustrates this point. Consider the infinitely repeated game with average reward (i.e., where the payoff to a given agent is the limit average of his payoffs in the individual stage games, as in Definition 6.1.1), in which the stage game is the normal-form game shown in Figure 7.1.

Stackelberg game

First note that player 1 (the row player) has a dominant strategy, namely B. Also note that (B, L) is the unique Nash equilibrium of the game. Indeed, if player 1 were to play B repeatedly, it is reasonable to expect that player 2 would always respond with L. Of course, if player 1 were to choose T instead, then player 2's best response would be R, yielding player 1 a payoff of 3 which is greater than player 1's Nash equilibrium payoff. In a single-stage game it would be hard for player 1 to convince player 2 that he (player 1) will play T, since it is a strictly dominated strategy.[1] However, in a repeated-game setting agent 1 has an opportunity to put his payoff where his mouth is, and adopt the role of a teacher. That is, player 1 could repeatedly play T; presumably, after a while player 2, if he has any sense at all, would get the message and start responding with R.

In the preceding example it is pretty clear who the natural candidate for adopting the teacher role is. But consider now the repetition of the Coordination game, reproduced in Figure 7.2. In this case, either player could play the teacher with equal success. However, if both decide to play teacher and happen to select uncoordinated actions (Left, Right) or (Right, Left) then the players will receive a payoff of zero forever.[2] Is there a learning rule that will enable them to coordinate without an external designation of a teacher?

7.1.2 *What constitutes learning?*

In the preceding examples the setting was a repeated game. We consider this a "learning" setting because of the temporal nature of the domain, and the regularity across time (at each time the same players are involved, and they play the same game as before). This allows us to consider strategies in which future action is

1. See related discussion on signaling and cheap talk in Chapter 8.
2. This is reminiscent of the "sidewalk shuffle," that awkward process of trying to get by the person walking toward you while he is doing the same thing, the result being that you keep blocking each other.

	Left	Right
Left	1, 1	0, 0
Right	0, 0	1, 1

Figure 7.2 Who's the teacher here?

selected based on the experience gained so far. When discussing repeated games in Chapter 6 we mentioned a few simple strategies. For example, in the context of repeated Prisoner's Dilemma, we mentioned the Tit-for-Tat (TfT) and trigger strategies. These, in particular TfT, can be viewed as very rudimentary forms of learning strategies. But one can imagine much more complex strategies, in which an agent's next choice depends on the history of play in more sophisticated ways. For example, the agent could guess that the frequency of actions played by his opponent in the past might be his current mixed strategy, and play a best response to that mixed strategy. As we shall see in Section 7.2, this basic learning rule is called *fictitious play*.

Repeated games are not the only context in which learning takes place. Certainly the more general category of stochastic games (also discussed in Chapter 6) is also one in which regularity across time allows meaningful discussion of learning. Indeed, most of the techniques discussed in the context of repeated games are applicable more generally to stochastic games, though specific results obtained for repeated games do not always generalize.

In both cases—repeated and stochastic games—there are additional aspects of the settings worth discussing. These have to do with whether the (e.g., repeated) game is commonly known by the players. If it is, any "learning" that takes place is only about the strategies employed by the other. If the game is not known, the agent can in addition learn about the structure of the game itself. For example, in a stochastic game setting, the agent may start out not knowing the payoff functions at a given stage game or the transition probabilities, but learn those over time in the course of playing the game. It is most interesting to consider the case in which the game being played is unknown; in this case there is a genuine process of discovery going on. (Such a setting could be modeled as a Bayesian game, as described in Section 6.3, though the formal modeling details are not necessary for the discussion in this chapter.) Some of the remarkable results are that, with certain learning strategies, agents can sometimes converge to an equilibrium of the game even without knowing the game being played. Additionally, there is the question of whether the game is *observable*; do the players see each others' actions, and/or each others' payoffs? (Of course, in the case of a known game, the actions also reveal the payoffs.)

observability

While repeated and stochastic games constitute the main setting in which we will investigate learning, there are other settings as well. Chief among them are models of large populations. These models, which were largely inspired by

	Yield	Dare
Yield	2, 2	1, 3
Dare	3, 1	0, 0

Figure 7.3 The game of Chicken.

evolutionary models in biology, are superficially quite different from the setting of repeated or stochastic games. Unlike the latter, which involve a small number of players, the evolutionary models consist of a large number of players, who repeatedly play a given game among themselves (e.g., pairwise in the case of two-player games). A closer look, however, shows that these models are in fact closely related to the models of repeated games. We discuss this further in the last section of this chapter.

7.1.3 *If learning is the answer, what is the question?*

It is very important to be clear on why we study learning in multiagent systems, and how we judge whether a given learning theory is successful or not. These might seem like trivial questions, but in fact the answers are not obvious and not unique.

First, note that in the following, when we speak about learning strategies, these should be understood as complete strategies, which involve learning in the sense of choosing action as well as updating beliefs. One consequence is that learning in the sense of "accumulated knowledge" is not always beneficial. In the abstract, accumulating knowledge never hurts, since one can always ignore what has been learned. But when one precommits to a particular strategy for acting on accumulated knowledge, sometimes less is more.

This point is related to the inseparability of learning from teaching, discussed earlier. For example, consider a protagonist agent planning to play an infinitely repeated game of *Chicken*, depicted in Figure 7.3. In the presence of any opponent who attempts to learn the protagonist agent's strategy and play a best response, an optimal strategy is to play the stationary policy of always daring; this is the "watch out: I'm crazy" policy. The opponent will learn to always yield, a worse outcome for him than learning anything.[3]

Chicken game

Broadly speaking, we can divide theories of learning in multiagent systems into two categories—*descriptive theories* and *prescriptive theories*.

descriptive theory

prescriptive theory

3. The literary-minded reader may be reminded of the quote from Oscar Wilde's *A Woman of No Importance*: "[...] the worst tyranny the world has ever known; the tyranny of the weak over the strong. It is the only tyranny that ever lasts." Except here it is the tyranny of the simpleton over the sophisticated.

Descriptive theories

Descriptive theories attempt to study the way learning takes place in real life—usually by people, but sometimes by other entities such as organizations or animal species. The goal here is to show experimentally that a certain model of learning agrees with behavior (typically, in laboratory experiments) and then to identify interesting properties of the formal model.

The ideal descriptive theory would have two properties.

realism **Property 7.1.1 (Realism)** *There should be a good match between the formal theory and the natural phenomenon being studied.*

convergence **Property 7.1.2 (Convergence)** *The formal theory should exhibit interesting behavioral properties, in particular convergence of the strategy profile being played to some solution concept (e.g., equilibrium) of the game being played.*

One approach to demonstrating realism is to apply the experimental methodology of the social sciences. While we will not focus on this approach, there are several good examples of it in economics and game theory. But there can be other reasons for studying a given learning process. For example, to the extent that one accepts the Bayesian model as at least an idealized model of human decision making, this model provides support for the idea of *rational learning*, which we discuss later.

Convergence properties come in various flavors. Here we survey four of them.

First of all, the holy grail has been showing convergence to stationary strategies which form a Nash equilibrium of the stage game. In fact often this is the hidden motive of the research. It has been noted that game theory is somewhat unusual in having the notion of an equilibrium without associated dynamics that give rise to the equilibrium. Showing that the equilibrium arises naturally would correct this anomaly.[4]

A second approach recognizes that actual convergence to Nash equilibria is a rare occurrence under many learning processes. It pursues an alternative: not requiring that the agents converge to a strategy profile that is a Nash equilibrium, but rather requiring that the empirical frequency of play converge to such an equilibrium. For example, consider a repeated game of Matching Pennies. If both agents repeatedly played (H,H) and (T,T), the frequency of both their plays would converge to (.5, .5), the strategy in the unique Nash equilibrium, even though the payoffs obtained would be very different from the equilibrium payoffs.

Third and yet more radically, we can give up entirely on Nash equilibrium as the relevant solution concept. One alternative is to seek convergence to a *correlated* equilibrium of the stage game. This is interesting in a number of ways. No-regret learning, which we discuss later, can be shown to converge to correlated equilibria in certain cases. Indeed, convergence to a correlated equilibrium provides a

4. However, recent theoretical progress on the complexity of computing a Nash equilibrium (see Section 4.2.1) raises doubts about whether any such procedure could be guaranteed to converge to an equilibrium, at least within polynomial time.

justification for the no-regret learning concept; the "correlating device" in this case is not an abstract notion, but the prior history of play.

Finally, we can give up on convergence to stationary policies, but require that the non-stationary policies converge to an interesting state. In particular, learning strategies that include building an explicit model of the opponents' strategies (as we shall see, these are called *model-based* learning rules) can be required to converge to correct models of the opponents' strategies.

Prescriptive theories

In contrast with descriptive theories, prescriptive theories ask how agents—people, programs, or otherwise—*should* learn. A such they are not required to show a match with real-world phenomena. By the same token, their main focus is not on behavioral properties, though they may investigate convergence issues as well. For the most part, we will concentrate on *strategic* normative theories, in which individual agents are self-motivated.

In zero-sum games, and even in repeated or stochastic zero sum games, it is meaningful to ask whether an agent is learning in an optimal fashion. But in general this question is not meaningful, since the answer depends not only on the learning being done but also on the behavior of other agents in the system. When all agents adopt the same strategy (e.g., they all adopt TfT, or all adopt reinforcement learning, to be discussed shortly), this is called *self-play*. One way to judge learning procedures is based on their performance in self-play. However, learning agents can be judged also by how they do in the context of other types of agents; a TfT agent may perform well against another TfT agent, but less well against an agent using reinforcement learning.

self-play

No learning procedure is optimal against all possible opponent behaviors. This observation is simply an instance of the general move in game theory away from the notion of "optimal strategy" and toward "best response" and equilibrium. Indeed, in the broad sense in which we use the term, a "learning strategy" is simply a strategy in a game that has a particular structure (namely, the structure of a repeated or stochastic game) that happens to have a component that is naturally viewed as adaptive.

So how do we evaluate a prescriptive learning strategy? There are several answers. The first is to adopt the standard game-theoretic stance: give up on judging a strategy in isolation, and instead ask which learning rules are in equilibrium with each other. Note that requiring that repeated-game learning strategies be in equilibrium with each other is very different from the convergence requirements discussed above; those speak about equilibrium in the stage game, not in the repeated game. For example, TfT is in equilibrium with itself in an infinitely repeated Prisoner's Dilemma game, but does not lead to the repeated Defect play, the only Nash equilibrium of the stage game. This "equilibrium of learning strategies" approach is not common, but we shall see one example of it later on.

A more modest, but by far more common and perhaps more practical approach is to ask whether a learning strategy achieves payoffs that are "high enough." This approach is both stronger and weaker than the requirement of "best response."

Best response requires that the strategy yield the highest possible payoff against a particular strategy of the opponent(s). A focus on "high enough" payoffs can consider a broader class of opponents, but makes weaker requirements regarding the payoffs, which are allowed to fall short of best response.

There are several different versions of such high-payoff requirements, each adopting and/or combining different basic properties.

safety of a learning rule

Property 7.1.3 (Safety) *A learning rule is safe if it guarantees the agent at least its maxmin payoff, or "security value." (Recall that this is the payoff the agent can guarantee to himself regardless of the strategies adopted by the opponents; see Definition 3.4.1.)*

rationality of a learning rule

Property 7.1.4 (Rationality) *A learning rule is rational if whenever the opponent settles on a stationary strategy of the stage game (i.e., the opponent adopts the same mixed strategy each time, regardless of the past), the agent settles on a best response to that strategy.*

universal consistency

Hannan consistency

no-regret

Property 7.1.5 (No-regret, informal) *A learning rule is* universally consistent, *or* Hannan consistent, *or exhibits* no regret *(these are all synonymous terms), if, loosely speaking, against any set of opponents it yields a payoff that is no less than the payoff the agent could have obtained by playing any one of his pure strategies throughout. We give a more formal definition of this condition later in the chapter.*

Some of these basic requirements are quite strong, and can be weakened in a variety of ways. One way is to allow slight deviations, either in terms of the magnitude of the payoff obtained, or the probability of obtaining it, or both. For example, rather than require optimality, one can require ϵ, δ-optimality, meaning that with probability of at least $1 - \delta$ the agent's payoff comes within ϵ of the payoff obtained by the best response. Another way of weakening the requirements is to limit the class of opponents against which the requirement holds. For example, attention can be restricted to the case of self play, in which the agent plays a copy of itself. (Note that while the learning strategies are identical, the game being played may not be symmetric.) For example, one might require that the learning rule guarantee convergence in self play. More broadly, as in the case of *targeted optimality*, which we discuss later, one might require a best response only against a particular class of opponents.

In the next sections, as we discuss several learning rules, we will encounter various versions of these requirements and their combinations. For the most part we will concentrate on repeated, two-player games, though in some cases we will broaden the discussion and discuss stochastic games and games with more than two players.

7.2 Fictitious play

fictitious play

Fictitious play is one of the earliest learning rules. It was actually not proposed initially as a learning model at all, but rather as an iterative method for computing

Nash equilibria in zero-sum games. It happens to not be a particularly effective way of performing this computation, but since it employs an intuitive update rule, it is usually viewed as a model of learning, albeit a simplistic one, and subjected to convergence analyses of the sort discussed above.

Fictitious play is an instance of model-based learning, in which the learner explicitly maintains beliefs about the opponent's strategy. The structure of such techniques is straightforward.

Initialize beliefs about the opponent's strategy
repeat
  Play a best response to the assessed strategy of the opponent
  Observe the opponent's actual play and update beliefs accordingly

Note that in this scheme the agent is oblivious to the payoffs obtained or obtainable by other agents. We do however assume that the agent knows his own payoff matrix in the stage game (i.e., the payoff he would get in each action profile, whether or not encountered in the past).

In fictitious play, an agent believes that his opponent is playing the mixed strategy given by the empirical distribution of the opponent's previous actions. That is, if A is the set of the opponent's actions, and for every $a \in A$ we let $w(a)$ be the number of times that the opponent has played action a, then the agent assesses the probability of a in the opponent's mixed strategy as

$$P(a) = \frac{w(a)}{\sum_{a' \in A} w(a')}.$$

For example, in a repeated Prisoner's Dilemma game, if the opponent has played C, C, D, C, D in the first five games, before the sixth game he is assumed to be playing the mixed strategy $(0.6, 0.4)$. Note that we can represent a player's beliefs with either a probability measure or with the set of counts $(w(a_1), \ldots, w(a_k))$.

We have not fully specified fictitious play. There exist different versions of fictitious play which differ on the tie-breaking method used to select an action when there is more than one best response to the particular mixed strategy induced by an agent's beliefs. In general the tie-breaking rule chosen has little effect on the results of fictitious play.

On the other hand, fictitious play is very sensitive to the players' initial beliefs. This choice, which can be interpreted as action counts that were observed before the start of the game, can have a radical impact on the learning process. Note that one must pick some nonempty prior belief for each agent; the prior beliefs cannot be $(0, \ldots, 0)$ since this does not define a meaningful mixed strategy.

Fictitious play is somewhat paradoxical in that each agent assumes a stationary policy of the opponent, yet no agent plays a stationary policy except when the process happens to converge to one. The following example illustrates the operation of fictitious play. Recall the Matching Pennies game from Chapter 3, reproduced here as Figure 7.4. Two players are playing a repeated game of Matching Pennies. Each player is using the fictitious play learning rule to update

	Heads	Tails
Heads	1, −1	−1, 1
Tails	−1, 1	1, −1

Figure 7.4 Matching Pennies game.

Round	1's action	2's action	1's beliefs	2's beliefs
0			(1.5,2)	(2,1.5)
1	T	T	(1.5,3)	(2,2.5)
2	T	H	(2.5,3)	(2,3.5)
3	T	H	(3.5,3)	(2,4.5)
4	H	H	(4.5,3)	(3,4.5)
5	H	H	(5.5,3)	(4,4.5)
6	H	H	(6.5,3)	(5,4.5)
7	H	T	(6.5,4)	(6,4.5)
⋮	⋮	⋮	⋮	⋮

Table 7.1 Fictitious play of a repeated game of Matching Pennies.

his beliefs and select actions. Player 1 begins the game with the prior belief that player 2 has played heads 1.5 times and tails 2 times. Player 2 begins with the prior belief that player 1 has played heads 2 times and tails 1.5 times. How will the players play?

The first seven rounds of play of the game is shown in Table 7.1.

As you can see, each player ends up alternating back and forth between playing heads and tails. In fact, as the number of rounds tends to infinity, the empirical distribution of the play of each player will converge to $(0.5, 0.5)$. If we take this distribution to be the mixed strategy of each player, the play converges to the unique Nash equilibrium of the normal form stage game, that in which each player plays the mixed strategy $(0.5, 0.5)$.

Fictitious play has several nice properties. First, connections can be shown to pure-strategy Nash equilibria, when they exist.

steady state

absorbing state

Definition 7.2.1 (Steady state) *An action profile a is a* steady state *(or* absorbing state*) of fictitious play if it is the case that whenever a is played at round t it is also played at round $t + 1$ (and hence in all future rounds as well).*

The following two theorems establish a tight connection between steady states and pure-strategy Nash equilibria.

Theorem 7.2.2 *If a pure-strategy profile is a strict Nash equilibrium of a stage game, then it is a steady state of fictitious play in the repeated game.*

Note that the pure-strategy profile must be a *strict* Nash equilibrium, which means that no agent can deviate to another action without strictly decreasing its payoff. We also have a converse result.

Theorem 7.2.3 *If a pure-strategy profile is a steady state of fictitious play in the repeated game, then it is a (possibly weak) Nash equilibrium in the stage game.*

Of course, one cannot guarantee that fictitious play always converges to a Nash equilibrium, if only because agents can only play pure strategies and a pure-strategy Nash equilibrium may not exist in a given game. However, while the stage game strategies may not converge, the empirical distribution of the stage game strategies over multiple iterations may. And indeed this was the case in the Matching Pennies example given earlier, where the empirical distribution of the each player's strategy converged to their mixed strategy in the (unique) Nash equilibrium of the game. The following theorem shows that this was no accident.

Theorem 7.2.4 *If the empirical distribution of each player's strategies converges in fictitious play, then it converges to a Nash equilibrium.*

This seems like a powerful result. However, notice that although the theorem gives sufficient conditions for the empirical distribution of the players' actions to converge to a mixed-strategy equilibrium, we have not made any claims about the distribution of the particular outcomes played.

To better understand this point, consider the following example. Consider the *Anti-Coordination game* shown in Figure 7.5.

Anti-Coordination game

	A	B
A	0, 0	1, 1
B	1, 1	0, 0

Figure 7.5 The Anti-Coordination game.

Clearly there are two pure Nash equilibria of this game, (A, B) and (B, A), and one mixed Nash equilibrium, in which each agent mixes A and B with probability 0.5. Either of the two pure-strategy equilibria earns each player a payoff of 1, and the mixed-strategy equilibrium earns each player a payoff of 0.5.

Now let us see what happens when we have agents play the repeated Anti-Coordination game using fictitious play. Let us assume that the weight function for each player is initialized to (1, 0.5). The play of the first few rounds is shown in Table 7.2.

As you can see, the play of each player converges to the mixed strategy (0.5, 0.5), which is the mixed strategy Nash equilibrium. However, the payoff received by each player is 0, since the players never hit the outcomes with positive payoff. Thus, although the empirical distribution of the strategies converges to

Round	1's action	2's action	1's beliefs	2's beliefs
0			(1,0.5)	(1,0.5)
1	B	B	(1,1.5)	(1,1.5)
2	A	A	(2,1.5)	(2,1.5)
3	B	B	(2,2.5)	(2,2.5)
4	A	A	(3,2.5)	(3,2.5)
⋮	⋮	⋮	⋮	⋮

Table 7.2 Fictitious play of a repeated Anti-Coordination game.

	Rock	Paper	Scissors
Rock	0, 0	0, 1	1, 0
Paper	1, 0	0, 0	0, 1
Scissors	0, 1	1, 0	0, 0

Figure 7.6 Shapley's Almost-Rock-Paper-Scissors game.

the mixed strategy Nash equilibrium, the players may not receive the expected payoff of the Nash equilibrium, because their actions are miscorrelated.

Finally, the empirical distributions of players' actions need not converge at all. Consider the game in Figure 7.6. Note that this example, due to Shapley, is a modification of the rock-paper-scissors game; this game is not constant sum.

The unique Nash equilibrium of this game is for each player to play the mixed strategy $(1/3, 1/3, 1/3)$. However, consider the fictitious play of the game when player 1's weight function has been initialized to $(0, 0, 0.5)$ and player 2's weight function has been initialized to $(0, 0.5, 0)$. The play of this game is shown in Table 7.3. Although it is not obvious from these first few rounds, it can be shown that the empirical play of this game never converges to any fixed distribution.

For certain restricted classes of games we *are* guaranteed to reach convergence.

Theorem 7.2.5 *Each of the following is a sufficient condition for the empirical frequencies of play to converge in fictitious play:*

- *The game is zero sum;*
- *The game is solvable by iterated elimination of strictly dominated strategies;*
- *The game is a potential game;*[5]
- *The game is* $2 \times n$ *and has generic payoffs.*[6]

5. Actually an even more more general condition applies here, that the players have "identical interests," but we will not discuss this further here.

6. Full discussion of genericity in games lies outside the scope of this book, but here is the essential idea, at least for games in normal form. Roughly speaking, a game in normal form is generic if it does

Round	1's action	2's action	1's beliefs	2's beliefs
0			(0,0,0.5)	(0,0.5,0)
1	Rock	Scissors	(0,0,1.5)	(1,0.5,0)
2	Rock	Paper	(0,1,1.5)	(2,0.5,0)
3	Rock	Paper	(0,2,1.5)	(3,0.5,0)
4	Scissors	Paper	(0,3,1.5)	(3,0.5,1)
5	Scissors	Paper	(0,1.5,0)	(1,0,0.5)
⋮	⋮	⋮	⋮	⋮

Table 7.3 Fictitious play of a repeated game of the Almost-Rock-Paper-Scissors game.

Overall, fictitious play is an interesting model of learning in multiagent systems not because it is realistic or because it provides strong guarantees, but because it is very simple to state and gives rise to nontrivial properties. But it is very limited; its model of beliefs and belief update is mathematically constraining, and is clearly implausible as a model of human learning. There exist various variants of fictitious play that score somewhat better on both fronts. We will mention one of them—called *smooth fictitious play*—when we discuss no-regret learning methods.

7.3 Rational learning

rational learning

Bayesian learning

Rational learning (also sometimes called *Bayesian learning*) adopts the same general model-based scheme as fictitious play. Unlike fictitious play, however, it allows players to have a much richer set of beliefs about opponents' strategies. First, the set of strategies of the opponent can include repeated-game strategies such as TfT in the Prisoner's Dilemma game, not only repeated stage-game strategies. Second, the beliefs of each player about his opponent's strategies may be expressed by any probability distribution over the set of all possible strategies.

Bayesian updating

As in fictitious play, each player begins the game with some prior beliefs. After each round, the player uses *Bayesian updating* to update these beliefs. Let S be the set of the opponent's strategies considered possible by player i, and H be the set of possible histories of the game. Then we can use Bayes' rule to express the probability assigned by player i to the event in which the opponent is playing a particular strategy $s \in S$ given the observation of history $h \in H$, as

$$P_i(s|h) = \frac{P_i(h|s)P_i(s)}{\sum_{s' \in S} P_i(h|s')P_i(s')}.$$

not have any interesting property that does not also hold with probability 1 when the payoffs are selected independently from a sufficiently rich distribution (e.g., the uniform distribution over a fixed interval). Of course, to make this precise we would need to define "interesting" and "sufficiently." Intuitively, though, this means that the payoffs do not have accidental properties. A game whose payoffs are all distinct is necessarily generic.

	C	D
C	3, 3	0, 4
D	4, 0	1, 1

Figure 7.7 Prisoner's Dilemma game.

For example, consider two players playing the infinitely repeated Prisoner's Dilemma game, reproduced in Figure 7.7.

Suppose that the support of the prior belief of each player (i.e., the strategies of the opponent to which the player ascribes nonzero probability; see Definition 3.2.6) consists of the strategies $g_1, g_2, \ldots g_\infty$, defined as follows. g_∞ is the *trigger strategy* that was presented in Section 6.1.2. A player using the trigger strategy begins the repeated game by cooperating, and if his opponent defects in any round, he defects in every subsequent round. For $T < \infty$, g_T coincides with g_∞ at all histories shorter than T but prescribes unprovoked defection starting from time T on. Following this convention, strategy g_0 is the strategy of constant defection.

trigger strategy

Suppose furthermore that each player happens indeed to select a best response from among $g_0, g_1, \ldots, g_\infty$. (There are of course infinitely many additional best responses outside this set.) Thus each round of the game will be played according to some strategy profile (g_{T_1}, g_{T_2}).

After playing each round of the repeated game, each player performs Bayesian updating. For example, if player i has observed that player j has always cooperated, the Bayesian updating after history $h_t \in H$ of length t reduces to

$$P_i(g_T|h_t) = \begin{cases} 0 & \text{if } T \leq t; \\ \frac{P_i(g_T)}{\sum_{k=t+1}^{\infty} P_i(g_k)} & \text{if } T > t. \end{cases}$$

Rational learning is a very intuitive model of learning, but its analysis is quite involved. The formal analysis focuses on self-play, that is, on properties of the repeated game in which all agents employ rational learning (though they may start with different priors). Broadly, the highlights of this model are as follows:

- Under some conditions, in self-play rational learning results in agents having close to correct beliefs about the observable portion of their opponent's strategy.
- Under some conditions, in self-play rational learning causes the agents to converge toward a Nash equilibrium with high probability.
- Chief among these "conditions" *absolute continuity*, a strong assumption.

In the remainder of this section we discuss these points in more detail, starting with the notion of absolute continuity.

Definition 7.3.1 (Absolute continuity) *Let X be a set and let $\mu, \mu' \in \Pi(X)$ be probability distributions over X. Then the distribution μ is said to be* absolutely continuous *with respect to the distribution μ' iff for $x \subset X$ that is measurable*[7] *it is the case that if $\mu(x) > 0$ then $\mu'(x) > 0$.*

absolute continuity

Note that the players' beliefs and the actual strategies each induce probability distributions over the set of histories H. Let $s = (s_1, \ldots, s_n)$ be a strategy profile. If we assume that these strategies are used by the players, we can calculate the probability of each history of the game occurring, thus inducing a distribution over H. We can also induce such a distribution with a player's beliefs about players' strategies. Let S_j^i be a set of strategies that i believes possible for j, and $P_j^i \in \Pi(S_j^i)$ be the distribution over S_j^i believed by player i. Let $P_i = (P_1^i, \ldots, P_n^i)$ be the tuple of beliefs about the possible strategies of every player. Now, if player i assumes that all players (including himself) will play according to his beliefs, he can also calculate the probability of each history of the game occurring, thus inducing a distribution over H. The results that follow all require that the distribution over histories induced by the actual strategies is absolutely continuous with respect to the distribution induced by a player's beliefs; in other words, if there is a positive probability of some history given the actual strategies, then the player's beliefs should also assign the history positive probability. (Colloquially, it is sometimes said that the beliefs of the players must contain a *grain of truth*.) Although the results that follow are very elegant, it must be said that the absolute continuity assumption is a significant limitation of the theoretical results associated with rational learning.

grain of truth

In the Prisoner's Dilemma example discussed earlier, it is easy to see that the distribution of histories induced by the actual strategies is absolutely continuous with respect to the distribution predicted by the prior beliefs of the players. All positive probability histories in the game are assigned positive probability by the original beliefs of both players: if the true strategies are g_{T_1}, g_{T_2}, players assign positive probability to the history with cooperation up to time $t < \min(T_1, T_2)$ and defection in all times exceeding the $\min(T_1, T_2)$.

The rational learning model is interesting because it has some very desirable properties. Roughly speaking, players satisfying the assumptions of the rational learning model will have beliefs about the play of the other players that converge to the truth, and furthermore, players will in finite time converge to play that is arbitrarily close to the Nash equilibrium. Before we can state these results we need to define a measure of the similarity of two probability measures.

Definition 7.3.2 (ϵ-closeness) *Given an $\epsilon > 0$ and two probability measures μ and μ' on the same space, we say that μ is ϵ-close to μ' if there is a measurable set Q satisfying:*

- *$\mu(Q)$ and $\mu'(Q)$ are each greater than $1 - \epsilon$; and*
- *for every measurable set $A \subseteq Q$, we have that*

$$(1 + \epsilon)\mu'(A) \geq \mu(A) \geq (1 - \epsilon)\mu'(A).$$

7. Recall that a probability distribution over a domain X does not necessarily give a value for all subsets of X, but only over some σ-algebra of X, the collection of measurable sets.

Now we can state a result about the accuracy of the beliefs of a player using rational learning.

Theorem 7.3.3 (Rational learning and belief accuracy) *Let s be a repeated-game strategy profile for a given n-player game*[8]*, and let $P = P_1, \ldots, P_n$ be a tuple of probability distributions over such strategy profiles (P_i is interpreted as player i's beliefs). Let μ_s and μ_P be the distributions over infinite game histories induced by the strategy profile s and the belief tuple P, respectively. If we have that:*

- *at each round, each player i plays a best response strategy given his beliefs P_i;*
- *after each round each player i updates P_i using Bayesian updating; and*
- *μ_s is absolutely continuous with respect to μ_{P_i},*

then for every $\epsilon > 0$ and for almost every history in the support of μ_s (i.e., every possible history given the actual strategy profile s), there is a time T such that for all $t \geq T$, the play μ_{P_i} predicted by the player i's beliefs is ϵ-close to the distribution of play μ_s predicted by the actual strategies.

Thus a player's beliefs will eventually converge to the truth if he is using Bayesian updating, is playing a best response strategy, and the play predicted by the other players' real strategies is absolutely continuous with respect to that predicted by his beliefs. In other words, he will correctly predict the on-path portions of the other players' strategies.

Note that this result does *not* state that players will learn the true strategy being played by their opponents. As stated earlier, there are an infinite number of possible strategies that their opponent could be playing, and each player begins with a prior distribution that assigns positive probability to only some subset of the possible strategies. Instead, players' beliefs will accurately predict the play of the game, and no claim is made about their accuracy in predicting the off-path portions of the opponents' strategies.

Consider again the two players playing the infinitely repeated Prisoner's Dilemma game, as described in the previous example. Let us verify that, as Theorem 7.3.3 dictates, the future play of this game will be correctly predicted by the players. If $T_1 < T_2$ then from time $T_1 + 1$ on, player 2's posterior beliefs will assign probability 1 to player 1's strategy, g_{T_1}. On the other hand, player 1 will never fully know player 2's strategy, but will know that $T_2 > T_1$. However, this is sufficient information to predict that player 2 will always choose to defect in the future.

A player's beliefs must converge to the truth even when his strategy space is incorrect (does not include the opponent's actual strategy), as long as they satisfy the absolute continuity assumption. Suppose, for instance, that player 1 is playing the trigger strategy g_∞, and player 2 is playing tit-for-tat, but that player 1 believes that player 2 is also playing the trigger strategy. Thus player 1's beliefs about player 2's strategy are incorrect. Nevertheless, his beliefs will correctly predict the future play of the game.

8. That is, a tuple of repeated-game strategies, one for each player.

We have so far spoken about the accuracy of beliefs in rational learning. The following theorem addresses convergence to equilibrium. Note that the conditions of this theorem are identical to those of Theorem 7.3.3, and that the definition refers to the concept of an ϵ-Nash equilibrium from Section 3.4.7, as well as to ϵ-closeness as defined earlier.

Theorem 7.3.4 (Rational Learning and Nash) *Let s be a repeated-game strategy profile for a given n-player game, and let $P = P_1, \ldots, P_n$ be a a tuple of probability distributions over such strategy profiles. Let μ_s and μ_P be the distributions over infinite game histories induced by the strategy profile s and the belief tuple P, respectively. If we have that:*

- *at each round, each player i plays a best response strategy given his beliefs P_i;*
- *after each round each player i updates P_i using Bayesian updating; and*
- *μ_s is absolutely continuous with respect to μ_{P_i},*

then for every $\epsilon > 0$ and for almost every history in the support of μ_s there is a time T such that for every $t \geq T$ there exists an ϵ-equilibrium s^* of the repeated game in which the play μ_{P_i} predicted by player i's beliefs is ϵ-close to the play μ_{s^*} of the equilibrium.

In other words, if utility-maximizing players start with individual subjective beliefs with respect to which the true strategies are absolutely continuous, then in the long run, their behavior must be essentially the same as a behavior described by an ϵ-Nash equilibrium.

Of course, the space of repeated-game equilibria is huge, which leaves open the question of which equilibrium will be reached. Here notice a certain self-fulfilling property: players' optimism can lead to high rewards, and likewise pessimism can lead to low rewards. For example, in a repeated Prisoner's Dilemma game, if both players begin believing that their opponent will likely play the TfT strategy, they each will tend to cooperate, leading to mutual cooperation. If, on the other hand, they each assign high prior probability to constant defection, or to the grim-trigger strategy, they will each tend to defect.

7.4 Reinforcement learning

reinforcement learning

In this section we look at multiagent extensions of learning in MDPs, that is, in single-agent stochastic games (see Appendix C for a review of MDP essentials). Unlike the first two learning techniques discussed, and with one exception discussed in section 7.4.4, *reinforcement learning* does not explicitly model the opponent's strategy. The specific family of techniques we look at are derived from the Q-learning algorithm for learning in unknown (single-agent) MDPs. Q-learning is described in the next section, after which we present its extension to zero-sum stochastic games. We then briefly discuss the difficulty in extending the methods to general-sum stochastic games.

7.4.1 *Learning in unknown MDPs*

First, consider (single-agent) MDPs. Value iteration, as described in Appendix C, assumes that the MDP is known. What if we do not know the rewards or transition probabilities of the MDP? It turns out that, if we always know what state we are in and the reward received in each iteration, we can still converge to the correct Q-values.

Q-learning **Definition 7.4.1 (Q-learning)** *Q-learning is the following procedure:*

Initialize the Q-function and V values (arbitrarily, for example)
repeat *until convergence*
  Observe the current state s_t.
  Select action a_t and and take it.
  Observe the reward $r(s_t, a_t)$
  Perform the following updates (and do not update any other Q-values):
  $Q_{t+1}(s_t, a_t) \leftarrow (1-\alpha)Q_t(s_t, a_t) + \alpha_t(r(s_t, a_t) + \beta V_t(s_{t+1}))$
  $V_{t+1}(s) \leftarrow \max_a Q_t(s, a)$

Theorem 7.4.2 *Q-learning guarantees that the Q and V values converge to those of the optimal policy, provided that each state-action pair is sampled an infinite number of times, and that the time-dependent learning rate α_t obeys $0 \leq \alpha_t < 1$, $\sum_0^\infty \alpha_t = \infty$ and $\sum_0^\infty \alpha_t^2 < \infty$.*

The intuition behind this approach is that we approximate the unknown transition probability by using the actual distribution of states reached in the game itself. Notice that this still leaves us a lot of room in designing the order in which the algorithm selects actions.

Note that this theorem says nothing about the rate of convergence. Furthermore, it gives no assurance regarding the accumulation of optimal future discounted rewards by the agent; it could well be, depending on the discount factor, that by the time the agent converges to the optimal policy it has paid too high a cost, which cannot be recouped by exploiting the policy going forward. This is not a concern if the learning takes place during training sessions, and only when learning has converged sufficiently is the agent unleashed on the world (e.g., think of a fighter pilot being trained on a simulator before going into combat). But in general Q-learning should be thought of as guaranteeing good learning, but neither quick learning nor high future discounted rewards.

7.4.2 *Reinforcement learning in zero-sum stochastic games*

In order to adapt the method presented from the setting of MDPs to stochastic games, we must make a few modifications. The simplest possible modification is to have each agent ignore the existence of the other agent (recall that zero-sum games involve only two agents). We then define $Q_i^\pi : S \times A_i \mapsto \mathbb{R}$ to be the value for player i if the two players follow strategy profile π after starting in state s and player i chooses the action a. We can now apply the Q-learning algorithm. As mentioned earlier in the chapter, the multiagent setting forces us to forego

our search for an "optimal" policy, and instead to focus on one that performs well against its opponent. For example, we might require that it satisfy Hannan consistency (Property 7.1.5). Indeed, the Q-learning procedure can be shown to be Hannan-consistent for an agent in a stochastic game against opponents playing stationary policies. However, against opponents using more complex strategies, such as Q-learning itself, we do not obtain such a guarantee.

The above approach, assuming away the opponent, seems unmotivated. Instead, if the agent is aware of what actions its opponent selected at each point in its history, we can use a modified Q-function, $Q_i^\pi : S \times A \mapsto \mathbb{R}$, defined over states and action profiles, where $A = A_1 \times A_2$. The formula to update Q is simple to modify and would be the following for a two-player game.

$$Q_{i,t+1}(s_t, a_t, o_t) = (1 - \alpha_t)Q_{i,t}(s_t, a_t, o_t) + \alpha_t(r_i(s_t, a_t, o_t) + \beta V_t(s_{t+1}))$$

Now that the actions range over both our agent's actions and that of its competitor, how can we calculate the value of a state? Recall that for (two-player) zero-sum games, the policy profile where each agent plays its maxmin strategy forms a Nash equilibrium. The payoff to the first agent (and thus the negative of the payoff to the second agent) is called the *value* of the game, and it forms the basis for our revised value function for Q-learning,

value of a zero-sum game

$$V_t(s) = \max_{\Pi_i} \min_{o} Q_{i,t}(s, \Pi_i(s), o).$$

minimax-Q

Like the basic Q-learning algorithm, the above *minimax-Q* learning algorithm is guaranteed to converge in the limit of infinite samples of each state and action profile pair. While this will guarantee the agent a payoff at least equal to that of its maxmin strategy, it no longer satisfies Hannan consistency. If the opponent is playing a suboptimal strategy, minimax-Q will be unable to exploit it in most games.

The minimax-Q algorithm is described in Figure 7.8. Note that this algorithm specifies not only how to update the Q and V values, but also how to update the strategy Π. There are still some free parameters, such as how to update the learning parameter, α. One way of doing so is to simply use a decay rate, so that α is set to $\alpha * decay$ after each Q-value update, for some value of $decay < 1$. Another possibility from the Q-learning literature is to keep separate α's for each state and action profile pair. In this case, a common method is to use $\alpha = 1/k$, where k equals the number of times that particular Q-value has been updated including the current one. So, when first encountering a reward for a state s where an action profile a was played, the Q-value is set entirely to the observed reward plus the discounted value of the successor state ($\alpha = 1$). On the next time that state–action profile pair is encountered, it will be set to be half of the old Q-value plus half of the new reward and discounted successor state value.

We now look at an example demonstrating the operation of minimax-Q learning in a simple repeated game: repeated Matching Pennies (see Figure 7.4) against an unknown opponent. Note that the convergence results for Q-learning impose only weak constraints on how to select actions and visit states. In this example, we follow the given algorithm and assume that the agent chooses an action randomly

// Initialize:
forall $s \in S, a \in A$, *and* $o \in O$ **do**
  $Q(s, a, o) \leftarrow 1$
forall s *in* S **do**
  $V(s) \leftarrow 1$
forall $s \in S$ *and* $a \in A$ **do**
  $\Pi(s, a) \leftarrow 1/|A|$
$\alpha \leftarrow 1.0$
// Take an action:
when in state s, with probability *explor* choose an action uniformly at random, and with probability $(1 - explor)$ choose action a with probability $\Pi(s, a)$
// Learn:
after receiving reward rew for moving from state s to s' via action a and opponent's action o
$Q(s, a, o) \leftarrow (1 - \alpha) * Q(s, a, o) + \alpha * (rew + \gamma * V(s'))$
$\Pi(s, \cdot) \leftarrow \arg\max_{\Pi'(s,\cdot)}(\min_{o'} \sum_{a'}(\Pi(s, a') * Q(s, a', o')))$
// The above can be done, for example, by linear programming
$V(s) \leftarrow \min_{o'}(\sum_{a'}(\Pi(s, a') * Q(s, a', o')))$
Update α

Figure 7.8 The minimax-Q algorithm.

some fraction of the time (denoted *explor*), and plays according to his current best strategy otherwise. For updating the learning rate, we have chosen the second method discussed earlier, with $\alpha = 1/k$, where k is the number of times the state and action profile pair has been observed. Assume that the Q-values are initialized to 1 and that the discount factor of the game is 0.9.

Table 7.4 shows the values of player 1's Q-function in the first few iterations of this game as well as his best strategy at each step. We see that the value of the game, 0, is being approached, albeit slowly. This is not an accident.

Theorem 7.4.3 *Under the same conditions that assure convergence of Q-learning to the optimal policy in MDPs, in zero-sum games Minimax-Q converges to the value of the game in self play.*

Here again, no guarantee is made about the rate of convergence or about the accumulation of optimal rewards. We can achieve more rapid convergence if we are willing to sacrifice the guarantee of finding a perfectly optimal maxmin strategy. In particular, we can consider the framework of *probably approximately correct (PAC) learning*. In this setting, choose some $\epsilon > 0$ and $1 > \delta > 0$, and seek an algorithm that can guarantee—regardless of the opponent—a payoff of at least that of the maxmin strategy minus ϵ, with probability $(1 - \delta)$. If we are willing to settle for this weaker guarantee, we gain the property that it will always hold after a polynomially-bounded number of time steps.

probably approximately correct (PAC) learning

One example of such an algorithm is the model-based learning algorithm *R-max*. It first initializes its estimate of the value of each state to be the highest

R-max algorithm

t	Actions	Reward$_1$	Q$_t$(H,H)	Q$_t$(H,T)	Q$_t$(T,H)	Q$_t$(T,T)	V(s)	π_1(H)
0			1	1	1	1	1	0.5
1	(H*,H)	1	1.9	1	1	1	1	0.5
2	(T,H)	-1	1.9	1	-0.1	1	1	0.55
3	(T,T)	1	1.9	1	-0.1	1.9	1.279	0.690
4	(H*,T)	-1	1.9	0.151	-0.1	1.9	0.967	0.534
5	(T,H)	-1	1.9	0.151	-0.115	1.9	0.964	0.535
6	(T,T)	1	1.9	0.151	-0.115	1.884	0.960	0.533
7	(T,H)	-1	1.9	0.151	-0.122	1.884	0.958	0.534
8	(H,T)	-1	1.9	0.007	-0.122	1.884	0.918	0.514
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
100	(H,H)	1	1.716	-0.269	-0.277	1.730	0.725	0.503
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1000	(T,T)	1	1.564	-0.426	-0.415	1.564	0.574	0.500
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 7.4 Minimax-Q learning in a repeated Matching Pennies game.

reward that can be returned in the game (hence the name). This philosophy has been referred to as *optimism in the face of uncertainty* and helps guarantee that the agent will explore its environment to the best of its ability. The agent then uses these optimistic values to calculate a maxmin strategy for the game. Unlike normal Q-learning, the algorithm does not update its values for any state and action profile pair until it has visited them "enough" times to have a good estimate of the reward and transition probabilities. Using a theoretical method called *Chernoff bounds*, it is possible to polynomially bound the number of samples necessary to guarantee that the accuracy of the average over the samples deviates from the true average by at most ϵ with probability $(1-\delta)$ for any selected value of ϵ and δ. The polynomial is in Σ, k, T, $1/\epsilon$, and $1/\delta$, where Σ is the number of states (or games) in the stochastic game, k is the number of actions available to each agent in a game (without loss of generally we can assume that this is the same for all agents and all games), and T is the ϵ-return *mixing time* of the optimal policy, that is, the smallest length of time after which the optimal policy is guaranteed to yield an expected payoff at most ϵ away from optimal. The notes at the end of the chapter point to further reading on R-max, and a predecessor algorithm called *E3* (pronounced "E cubed").

Chernoff bounds

mixing time

E3 algorithm

7.4.3 *Beyond zero-sum stochastic games*

So far we have shown results for the class of zero-sum stochastic games. Although the algorithms discussed, in particular minimax-Q, are still well defined in the general-sum case, the guarantee of achieving the maxmin strategy payoff is less compelling. Another subclass of stochastic games that has been addressed is that of common-payoff (pure coordination) games, in which all agents receive the same reward for an outcome. This class has the advantage of reducing the problem to identifying an optimal action profile and coordinating with the other agents to play it. In many ways this problem can really be seen as a single-agent

problem of distributed control. This is a relatively well-understood problem, and various algorithms exist for it, depending on precisely how the problem is defined.

Expanding reinforcement learning algorithms to the general-sum case is quite problematic, on the other hand. There have been attempts to generalize Q-learning to general-sum games, but they have not yet been truly successful. As was discussed at the beginning of this chapter, the question of what it means to learn in general-sum games is subtle. One yardstick we have discussed is convergence to Nash equilibrium of the stage game during self play. No generalization of Q-learning has been put forward that has this property.

7.4.4 *Belief-based reinforcement learning*

There is also a version of reinforcement learning that includes explicit modeling of the other agent(s), given by the following equations.

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\alpha)Q_t(s_t, a_t) + \alpha_t(r(s_t, a_t) + \beta V_t(s_{t+1}))$$

$$V_t(s) \leftarrow \max_{a_i} \sum_{a_{-i} \subset A_{-i}} Q_t(s, (a_i, a_{-i}))Pr_i(a_{-i})$$

In this version, the agent updates the value of the game using the probability he assigns to the opponent(s) playing each action profile. Of course, the belief function must be updated after each play. How it is updated depends on what the function is. Indeed, belief-based reinforcement learning is not a single procedure but a family, each member characterized by how beliefs are formed and updated. For example, in one version the beliefs are of the kind considered in fictitious play, and in another they are Bayesian in the style of rational learning. There are some experimental results that show convergence to equilibrium in self-play for some versions of belief-based reinforcement learning and some classes of games, but no theoretical results.

7.5 No-regret learning and universal consistency

As discussed above, learning rule is universally consistent or (equivalently) exhibits no regret if, loosely speaking, against any set of opponents it yields a payoff that is no less than the payoff the agent could have obtained by playing any one of his pure strategies throughout.

More precisely, let α^t be the average per-period reward the agent received up until time t, and let $\alpha^t(s_i)$ be the average per-period reward the agent *would have* received up until time t had he played pure strategy s instead, assuming all other agents continue to play as they did.

regret **Definition 7.5.1 (Regret)** *The* regret *an agent experiences at time t for not having played s is $R^t(s) = \alpha^t - \alpha^t(s)$.*

Observe that this is conceptually the same as the definition of regret we offered in Section 3.4 (Definition 3.4.5).

A learning rule is said to exhibit *no regret*[9] if it guarantees that with high probability the agent will experience no positive regret.

no-regret

Definition 7.5.2 (No-regret learning rule) *A learning rule exhibits* no regret *if for any pure strategy of the agent s it holds that* $Pr([\liminf R^t(s)] \leq 0) = 1$.

The quantification is over all of the agent's pure strategies of the stage game, but note that it would make no difference if instead one quantified over all mixed strategies of the stage game. (Do you see why?) Note also that this guarantee is only in expectation, since the agent's strategy will in general be mixed, and thus the payoff obtained at any given time—u_i^t—is uncertain.

It is important to realize that this "in hindsight" requirement ignores the possibility that the opponents' play might change as a result of the agent's own play. This is true for stationary opponents, and might be a reasonable approximation in the context of a large number of opponents (such as in a public securities market), but less in the context of a small number of agents, of the sort game theory tends to focus on. For example, in the finitely-repeated Prisoner's Dilemma game, the only strategy exhibiting no regret is to always defect. This precludes strategies that capitalize on cooperative behavior by the opponent, such as Tit-for-Tat. In this connection see our earlier discussion of the inseparability of learning and teaching.

regret matching

smooth fictitious play

Over the years, a variety of no-regret learning techniques have been developed. Here are two, *regret matching* and *smooth fictitious play*.

- *Regret matching*: At each time step each action is chosen with probability proportional to its regret. That is,

$$\sigma_i^{t+1}(s) = \frac{R^t(s)}{\sum_{s' \in S_i} R^t(s')},$$

  where $\sigma_i^{t+1}(s)$ is the probability that agent i plays pure strategy s at time $t+1$.
- *Smooth fictitious play*: Instead of playing the best response to the empirical frequency of the opponent's play, as fictitious play prescribes, one introduces a perturbation that gradually diminishes over time. That is, rather than adopt at time $t+1$ a pure strategy s_i that maximizes $u_i(s_i, P^t)$ where P^t is the empirical distribution of opponent's play until time t, agent i adopts a mixed strategy σ_i that maximizes $u_i(s_i, P^t) + \lambda v_i(\sigma_i)$. Here λ is any constant, and v_i is a smooth, concave function with boundaries at the unit simplex. For example, v_i can be the entropy function, $v_i(\sigma_i) = -\sum_{S_i} \sigma_i(s_i) \log \sigma_i(s_i)$.

Regret matching can be shown to exhibit no regret, and smooth fictitious play approaches no regret as λ tends to zero. The proofs are based on Blackwell's Approachability Theorem; the notes at the end of the chapter provide pointers for further reading on it, as well as on other no-regret techniques.

9. There are actually several versions of regret. The one described here is called *external regret* in computer science, and *unconditional regret* in game theory.

7.6 Targeted learning

No-regret learning was one approach to ensuring good rewards, but as we discussed this sense of "good" has some drawbacks. Here we discuss an alternative sense of "good," which retains the requirement of best response, but limits it to a particular class of opponents. The intuition guiding this approach is that in any strategic setting, in particular a multiagent learning setting, one has *some* sense of the agents in the environment. A chess player has studied previous plays of his opponent, a skipper in a sailing competition knows a lot about his competitors, and so on. And so it makes sense to try to optimize against this set of opponents, rather than against completely unknown opponents.

targeted learning

Technically speaking, the model of *targeted learning* takes as a parameter a class—the "target class"—of likely opponents and is required to perform particularly well against these likely opponents. At the same time one wants to ensure at least the maxmin payoff against opponents outside the target class. Finally, an additional desirable property is for the algorithm to perform well in self-play; the algorithm should be designed to "cooperate" with itself.

For games with only two agents, these intuitions can be stated formally as follows.

targeted optimality

Property 7.6.1 (Targeted optimality) *Against any opponent in the target class, the expected payoff is the best-response payoff.*[10]

safety

Property 7.6.2 (Safety) *Against any opponent, the expected payoff is at least the individual security (or maxmin) value for the game.*

autocompatibility

Property 7.6.3 (Autocompatibility) *Self-play—in which both agents adopt the learning procedure in question—is strictly Pareto efficient.*[11]

We introduce one additional twist. Since we are interested in quick learning, not only learning in the limit, we need to allow some departure from the ideal. And so we amend the requirements as follows.

efficient targeted learning

Definition 7.6.4 (Efficient targeted learning) *A learning rule exhibits* efficient targeted learning*if for every* $\epsilon > 0$ *and* $1 > \delta > 0$*, there exists an* M *polynomial in* $1/\epsilon$ *and* $1/\delta$ *such that after* M *time steps, with probability greater than* $1 - \delta$*, all three payoff requirements listed previously are achieved within* ϵ*.*

Note the difference from no-regret learning. For example, consider learning in a repeated Prisoner's Dilemma game. Suppose that the target class consists of all opponents whose strategies rely on the past iteration; note this includes the Tit-for-Tat strategy. In this case successful targeted learning will result in constant cooperation, while no-regret learning prescribes constant defection.

10. Note: the expectation is over the mixed-strategy profiles, but not over opponents; this requirement is for any fixed opponent.

11. Recall that strict Pareto efficiency means that one agent's expected payoff cannot increase without the other's decreasing; see Definition 3.3.2. Also note that we do not restrict the discussion to symmetric games, and so self play does not in general mean identical play by the agents, nor identical payoffs. We abbreviate "strictly Pareto efficient" as "Pareto efficient."

How hard is it to achieve efficient targeted learning? The answer depends of course on the target class. Provably correct (with respect to this criterion) learning procedures exist for the class of stationary opponents, and the class of opponents whose memory is limited to a finite window into the past. The basic approach is to construct a number of building blocks and then specialize and combine them differently depending on the precise setting. The details of the algorithms can get involved, especially in the interesting case of nonstationary opponents, but the essential flow is as follows.

1. Start by assuming that the opponent is in the target set and learn a best response to the particular agent under this assumption. If the payoffs you obtain stray too much from your expectation, move on.
2. Signal to the opponent to find out whether he is employing the same learning strategy. If he is, coordinate to a Pareto-efficient outcome. If your payoffs stray too far off, move on.
3. Play your security-level strategy.

Note that so far we have restricted the discussion to two-player games. Can we generalize the criteria—and the algorithms—to games with more players? The answer is yes, but various new subtleties creep in. For example, in the two-agent case we needed to worry about three cases, corresponding to whether the opponent is in the target set, is a self-play agent, or is neither. We must now consider three sets of agents—self play agents (i.e., agents using the algorithm in question), agents in the target set, and unconstrained agents, and ask how agents in the first set can jointly achieve a Pareto-efficient outcome against the second set and yet protect themselves from exploitation by agents in the third set. This raises questions about possible coordination among the agents:

- Can self-play agents coordinate other than implicitly through their actions?
- Can opponents—whether in the target set or outside—coordinate other than through the actions?

The section at the end of the chapter points to further reading on this topic.

7.7 Evolutionary learning and other large-population models

In this section we shift our focus from models of the learning of individual agents to models of the learning of populations of agents (although, as we shall see, we will not abandon the single-agent perspective altogether). When we speak about learning in a population of agents, we mean the change in the constitution and behavior of that population over time. These models were originally developed by population biologists to model the process of biological evolution, and later adopted and adapted by other fields.

In the first subsection we present the model of the *replicator dynamic*, a simple model inspired by evolutionary biology. In the second subsection we present the concept of *evolutionarily stable strategies*, a stability concept that is related to the replicator dynamic. We conclude with a somewhat different model of *agent-based simulation* and the concept of *emergent conventions*.

7.7.1 *The replicator dynamic*

replicator dynamic

The *replicator dynamic* models a population undergoing frequent interactions. We will concentrate on the symmetric, two-player case, in which the agents repeatedly play a two-player symmetric normal-form stage game[12] against each other.

symmetric game

Definition 7.7.1 (Symmetric 2×2 game) *Let a two-player two-action normal-form game be called a* symmetric game *if it has the following form:*

	A	B
A	x, x	u, v
B	v, u	y, y

Intuitively, this requirement says that the agents do not have distinct roles in the game, and the payoff for agents does not depend on their identities. We have already seen several instances of such games, including the Prisoner's Dilemma.[13]

The replicator dynamic describes a population of agents playing such a game in an ongoing fashion. At each point in time, each agent only plays a pure strategy. Informally speaking, the model then pairs all agents and has them play each other,

fitness

each obtaining some payoff. This payoff is called the agent's *fitness*. At this point the biological inspiration kicks in—each agent now "reproduces" in a manner proportional to this fitness, and the process repeats. The question is whether the process converges to a fixed proportion of the various pure strategies within the population, and if so to which fixed proportions.

The verbal description above is only meant to be suggestive. The actual mathematical model is a little different. First, we never explicitly model the play of the game between particular sets of players; we only model the proportions of the populations associated with a given strategy. Second, the model is not one of discrete repetitions of play, but rather one of continuous evolution. Third, beyond the fitness-based reproduction, there is also a random element that impacts the proportions in the population. (Again, because of the biological inspiration, this

mutation

random element is called *mutation*.)

The formal model is as follows. Given a normal-form game $G = (\{1, 2\}, A, u)$, let $\varphi_t(a)$ denote the number of players playing action a at time t. Also, let

$$\theta_t(a) = \frac{\varphi_t(a)}{\sum_{a' \in A} \varphi_t(a')}$$

12. There exist much more general notions of symmetric normal-form games with multiple actions and players, but the following is sufficient for our purposes.

13. This restriction to symmetric games is very convenient, simplifying both the substance and notation of what follows. However, there exist more complicated evolutionary models, including ones allowing both different strategy spaces for different agents and nonsymmetric payoffs. At the end of the chapter we point the reader to further reading on these models.

be the proportion of players playing action a at time t. We denote with φ_t the vector of measures of players playing each action, and with θ_t the vector of population shares for each action.

The expected payoff to any individual player for playing action a at time t is

$$u_t(a) = \sum_{a'} \theta_t(a')u(a, a').$$

The change in the number of agents playing action a at time t is defined to be proportional to his fitness, that is, his average payoff at the current time,

$$\dot{\varphi}_t(a) = \varphi_t(a)u_t(a).$$

The absolute numbers of agents of each type are not important; only the relative ratios are. Defining the average expected payoff of the whole population as

$$u_t^* = \sum_{a} \theta_t(a)u_t(a),$$

we have that the change in the fraction of agents playing action a at time t is

$$\dot{\theta}_t(a) = \frac{\left[\dot{\varphi}_t(a)\sum_{a'\in A}\varphi_t(a')\right] - \left[\varphi_t(a)\sum_{a'\in A}\dot{\varphi}_t(a')\right]}{\left[\sum_{a'\in A}\varphi_t(a')\right]^2} = \theta_t(a)[u_t(a) - u_t^*].$$

The system we have defined has a very intuitive quality. If an action does better than the population average then the proportion of the population playing this action increases, and vice versa. Note that even an action that is not a best response to the current population state can grow as a proportion of the population when its expected payoff is better than the population average.

How should we interpret this evolutionary model? A straightforward interpretation is that it describes agents repeatedly interacting and replicating within a large population. However, we can also interpret the fraction of agents playing a certain strategy as the mixed strategy of a single agent, and the process as that of two identical agents repeatedly updating their identical mixed strategies based on their previous interaction. Seen in this light, except for its continuous-time nature, the evolutionary model is not as different from the repeated-game model as it seems at first glance.

We would like to examine the equilibrium points in this system. Before we do, we need a definition of stability.

steady state

Definition 7.7.2 (Steady state) *A* steady state *of a population using the replicator dynamic is a population state θ such that for all $a \in A$, $\dot{\theta}(a) = 0$.*

In other words, a steady state is a state in which the population shares of each action are constant. This stability concept has a major flaw. Any state in which all players play the same action is a steady state. The population shares of the actions will remain constant because the replicator dynamic does not allow the "entry" of strategies that are not already being played. To disallow these states, we will often require that our steady states are *stable*.

stable steady state

Definition 7.7.3 (Stable steady state) *A steady state θ of a replicator dynamic is* stable *if for every neighborhood U of θ there is another neighborhood U' of θ such that if $\theta_0 \in U'$ then $\theta_t \in U$ for all $t > 0$.*

That is, if the system starts close enough to the steady state, it remains nearby.

Finally, we might like to define an equilibrium state which, if perturbed, will eventually return back to the state. We call this *asymptotic stability*.

asymptotically stable state

Definition 7.7.4 (Asymptotically stable state) *A steady state θ of a replicator dynamic is* asymptotically stable *if it is stable, and in addition if for every neighborhood U of θ it is the case that if $\theta_0 \in U$ then $\lim_{t\to\infty} \theta_t = \theta$.*

The following example illustrates some of these concepts. Consider a homogeneous population playing the Anti-Coordination game, repeated in Figure 7.9.

	A	B
A	0, 0	1, 1
B	1, 1	0, 0

Figure 7.9 The Anti-Coordination game.

The game has two pure-strategy Nash equilibria, (A, B) and (B, A), and one mixed-strategy equilibrium in which both players select actions from the distribution $(0.5, 0.5)$. Because of the symmetric nature of the setting, there is no way for the replicator dynamic to converge to the pure-strategy equilibria. However, note that the state corresponding to the mixed-strategy equilibrium is a steady state, because when half of the players are playing A and half are playing B, both strategies have equal expected payoff (0.5) and the population shares of each are constant. Moreover, notice that this state is also asymptotically stable. The replicator dynamic, when started in any other state of the population (where the share of players playing A is more or less than 0.5) will converge back to the state $(0.5, 0.5)$. More formally we can express this as

$$\begin{aligned}\dot{\theta}(A) &= \theta(A)(1 - \theta(A) - 2\theta(A)(1 - \theta(A))) \\ &= \theta(A)(1 - 3\theta(A) + 2\theta(A)^2).\end{aligned}$$

This expression is positive for $\theta(A) < 0.5$, exactly 0 at 0.5, and negative for $\theta(A) > 0.5$, implying that the state $(0.5, 0.5)$ is asymptotically stable.

This example suggests that there may be a special relationship between Nash equilibria and states in the replicator dynamic. Indeed, this is the case, as the following results indicate.

Theorem 7.5.5 *Given a normal-form game $G = (\{1, 2\}, A = \{a_1, \ldots, a_k\}, u)$, if the strategy profile (S, S) is a (symmetric) mixed strategy Nash equilibrium of G then the population share vector $\theta = (S(a_1), \ldots, S(a_k))$ is a steady state of the replicator dynamic of G.*

In other words, every symmetric Nash equilibrium is a steady state. The reason for this is quite simple. In a state corresponding to a mixed Nash equilibrium, all strategies being played have the same average payoff, so the population shares remain constant.

As mentioned above, however, it is not the case that every steady state of the replicator dynamic is a Nash equilibrium. In particular, states in which not all actions are played may be steady states because the replicator dynamic cannot introduce new actions, even when the corresponding mixed-strategy profile is not a Nash equilibrium. On the other hand, the relationship between Nash equilibria and *stable* steady states is much tighter.

Theorem 7.7.6 *Given a normal-form game* $G = (\{1, 2\}, A\{a_1, \ldots, a_k\}, u)$ *and a mixed strategy* S*, if the population share vector* $\theta = (S(a_1), \ldots, S(a_k))$ *is a stable steady state of the replicator dynamic of* G*, then the strategy profile* (S, S) *is a mixed strategy Nash equilibrium of* G*.*

In other words, every stable steady state is a Nash equilibrium. It is easier to understand the contrapositive of this statement. If a mixed-strategy profile is not a Nash equilibrium, then some action must have a higher payoff than some of the actions in its support. Then in the replicator dynamic the share of the population using this better action will increase, once it exists. Then it is not possible that the population state corresponding to this mixed-strategy profile is a stable steady state.

Finally, we show that asymptotic stability corresponds to a notion that is stronger than Nash equilibrium. Recall the definition of trembling-hand perfection (Definition 3.4.14), reproduced here for convenience.

Definition 7.7.7 (Trembling-hand perfect equilibrium) *A mixed strategy* S *is a* (trembling-hand) perfect equilibrium *of a normal-form game* G *if there exists a sequence* $S^0, S^1, \ldots$ *of fully mixed-strategy profiles such that* $\lim_{n\to\infty} S^n = S$*, and such that for each* S^k *in the sequence and each player* i*, the strategy* s_i *is a best response to the strategies* s^k_{-i}*.*

Furthermore, we say informally that an equilibrium strategy profile is *isolated* if there does not exist another equilibrium strategy profile in the neighborhood (i.e., reachable via small perturbations of the strategies) of the original profile. Then we can relate trembling-hand perfection to the replicator dynamic as follows.

Theorem 7.7.8 *Given a normal-form game* $G = (\{1, 2\}, A, u)$ *and a mixed strategy* S*, if the population share vector* $\theta = (S(a_1), \ldots, S(a_k))$ *is an asymptotically stable steady state of the replicator dynamic of* G*, then the strategy profile* (S, S) *is a Nash equilibrium of* G *that is trembling-hand perfect and isolated.*

7.7.2 *Evolutionarily stable strategies*

evolutionarily stable strategy (ESS)

An *evolutionarily stable strategy (ESS)* is a stability concept that was inspired by the replicator dynamic. However, unlike the steady states discussed earlier, it does not require the replicator dynamic, or any dynamic process, explicitly;

	H	D
H	$-2, -2$	$6, 0$
D	$0, 6$	$3, 3$

Figure 7.10 Hawk–Dove game.

rather it is a static solution concept. Thus in principle it is not inherently linked to learning.

Roughly speaking, an evolutionarily stable strategy is a mixed strategy that is "resistant to invasion" by new strategies. Suppose that a population of players is playing a particular mixed strategy in the replicator dynamic. Then suppose that a small population of "invaders" playing a different strategy is added to the population. The original strategy is considered to be an ESS if it gets a higher payoff against the resulting mixture of the new and old strategies than the invaders do, thereby "chasing out" the invaders.

More formally, we have the following.

Definition 7.7.9 (Evolutionarily stable strategy (ESS)) *Given a symmetric two-player normal-form game $G = (\{1, 2\}, A, u)$ and a mixed strategy S, we say that S is an evolutionarily stable strategy if and only if for some $\epsilon > 0$ and for all other strategies S' it is the case that*

$$u(S, (1 - \epsilon)S + \epsilon S') > u(S', (1 - \epsilon)S + \epsilon S').$$

We can use properties of expectation to state this condition equivalently as

$$(1 - \epsilon)u(S, S) + \epsilon u(S, S') > (1 - \epsilon)u(S', S) + \epsilon u(S', S').$$

Note that, since this only needs to hold for small ϵ, this is equivalent to requiring that either $u(S, S) > u(S', S)$ holds, or else both $u(S, S) = u(S', S)$ and $u(S, S') > u(S', S')$ hold. Note that this is a strict definition. We can also state a weaker definition of ESS.

weak evolutionarily stable strategy

Definition 7.7.10 (Weak ESS) *S is a* weak evolutionarily stable strategy *if and only if for some $\epsilon > 0$ and for all S' it is the case that either $u(S, S) > u(S', S)$ holds, or else both $u(S, S) = u(S', S)$ and $u(S, S') \geq u(S', S')$ hold.*

This weaker definition includes strategies in which the invader does just as well against the original population as it does against itself. In these cases the population using the invading strategy will not grow, but it will also not shrink.

We illustrate the concept of ESS with the instance of the *Hawk–Dove* game shown in Figure 7.10. The story behind this game might be as follows. Two animals are fighting over a prize such as a piece of food. Each animal can choose between two behaviors: an aggressive hawkish behavior H, or an accommodating dovish behavior D. The prize is worth 6 to each of them. Fighting costs each

player 5. When a hawk meets a dove he gets the prize without a fight, and hence the payoffs are 6 and 0, respectively. When two doves meet they split the prize without a fight, hence a payoff of 3 to each one. When two hawks meet a fight breaks out, costing each player 5 (or, equivalently, yielding -5). In addition, each player has a 50% chance of ending up with the prize, adding an expected benefit of 3, for an overall payoff of -2.

It is not hard to verify that the game has a unique symmetric Nash equilibrium (S, S), where $S = (\frac{3}{5}, \frac{2}{5})$, and that S is also the unique ESS of the game. To confirm that S is an ESS, we need that for all $S' \neq S$, $u(S, S) = u(S', S)$ and $u(S, S') > u(S', S')$. The equality condition is true of any mixed strategy equilibrium with full support, so follows directly. To demonstrate that the inequality holds, it is sufficient to find the S'—or equivalently, the probability of playing H—that minimizes $f(S') = u(S, S') - u(S', S')$. Expanding $f(S')$ we see that it is a quadratic equation with the (unique) maximum $S' = S$, proving our result.

This connection between an ESS and a Nash equilibrium is not accidental. The following two theorems capture this connection.

Theorem 7.7.11 *Given a symmetric two-player normal-form game* $G = (\{1, 2\}, A, u)$ *and a mixed strategy* S*, if* S *is an evolutionarily stable strategy then* (S, S) *is a Nash equilibrium of* G*.*

This is easy to show. Note that by definition an ESS S must satisfy

$$u(S, S) \geq u(S', S).$$

In other words, it is a best response to itself and thus must be a Nash equilibrium. However, not every Nash equilibrium is an ESS; this property is guaranteed only for strict equilibria.

Theorem 7.7.12 *Given a symmetric two-player normal-form game* $G = (\{1, 2\}, A, u)$ *and a mixed strategy* S*, if* (S, S) *is a strict (symmetric) Nash equilibrium of* G*, then* S *is an evolutionarily stable strategy.*

This is also easy to show. Note that for any strict Nash equilibrium S it must be the case that

$$u(S, S) > u(S', S).$$

But this satisfies the first criterion of an ESS.

The ESS also is related to the idea of stability in the replicator dynamic.

Theorem 7.7.13 *Given a symmetric two-player normal-form game* $G = (\{1, 2\}, A, u)$ *and a mixed strategy* S*, if* S *is an evolutionarily stable strategy then it is an asymptotically stable steady state of the replicator dynamic of* G*.*

Intuitively, if a state is an ESS then we know that it will be resistant to invasions by other strategies. Thus, when this strategy is represented by a population in the replicator dynamic, it will be resistant to small perturbations. What is interesting, however, is that the converse is *not* true. The reason for this is that in the replicator dynamic, only pure strategies can be inherited. Thus some states that are asymptotically stable would actually not be resistant to invasion by a mixed strategy, and thus not an ESS.

	A	B
A	x, x	u, v
B	v, u	y, y

Figure 7.11 A game for agent-based simulation models.

7.7.3 *Agent-based simulation and emergent conventions*

It was mentioned in Section 7.7.1 that, while motivated by a notion of dynamic process within a population, in fact the replicator dynamic only models the gross statistics of the process, not its details. There are other large-population models that provide a more fine-grained model of the process, with many parameters that can impact the dynamics. We call such models, which explicitly model the individual agents, *agent-based simulation* models.

agent-based simulation

In this we look at one such model, geared toward the investigation of how conventions emerge in a society. In Section 2.4 we saw how in any realistic multi-agent system it is crucial that the agents agree on certain *social laws*, in order to decrease conflicts among them and promote cooperative behavior. Without such laws even the simplest goals might become unattainable by any of the agents, or at least not efficiently attainable (just imagine driving in the absence of traffic rules). A social law restricts the options available to each agent. A special case of social laws are *social conventions*, which limit the agents to exactly one option from the many available ones (e.g., always driving on the right side of the road). A good social law or convention strike as balance between on the one hand allowing agents sufficient freedom to achieve their goals, and on the other hand restricting them so that they do not interfere too much with one another.

social law

social convention

In Section 2.4 we asked how social laws and conventions can be designed by a social designer, but here we ask how such conventions can emerge organically. Roughly speaking, the process we aim to study is one in which individual agents occasionally interact with one another, and as a result gain some new information. Based on his personal accumulated information, each agent updates his behavior over time. This process is reminiscent of the replicator dynamic, but there are crucial differences. We start in the same way, and restrict the discussion to symmetric, two-player-two-choices games. Here too one can look at much more general settings, but we will restrict ourselves to the game schema in Figure 7.11.

However, unlike the replicator dynamic, here we assume a discrete process, and furthermore assume that at each stage exactly one pair of agents—selected at random from the population—play. This contrasts sharply with the replicator dynamic, which can be interpreted as implicitly assuming that almost all pairs of agents play before updating their choices of action. In this discrete model each agent is tracked individually, and indeed different agents end up possessing very different information.

Most importantly, in contrast with the replicator dynamic, the evolution of the system is not defined by some global statistics of the system. Instead, each agent decides how to play the next game based on his individual accumulated experience thus far. There are two constraints we impose on such rules.

anonymous learning rule

Property 7.7.14 (Anonymity) *The selection function cannot be based on the identities of agents or the names of actions.*

local learning rule

Property 7.7.15 (Locality) *The selection function is purely a function of the agent's personal history; in particular, it is not a function of global system properties.*

The requirement of anonymity deserves some discussion. We are interested in how social conventions emerge when we cannot anticipate in advance the games that will be played. For example, if we know that the coordination problem will be that of deciding whether to drive on the left of the road or on the right, we can very well use the names "left" and "right" in the action-selection rule; in particular, we can admit the trivial update rule that has all agents drive on the right immediately. Instead, the type of coordination problem we are concerned with is better typified by the following example. Consider a collection of manufacturing robots that have been operating at a plant for five years, at which time a new collection of parts arrive that must be assembled. The assembly requires using one of two available attachment widgets, which were introduced three years ago (and hence were unknown to the designer of the robots five years ago). Either of the widgets will do, but if two robots use different ones then they incur the high cost of conversion when it is time for them to mate their respective parts. Our goal is that the robots learn to use the same kind of widget. The point to emphasize about this example is that five years ago the designer could have stated rules of the general form "if in the future you have several choices, each of which has been tried this many times and has yielded this much payoff, then next time make the following choice"; the designer could not, however, have referred to the specific choices of widget, since those were only invented two years later.

The prohibition on using agent identities in the rules (e.g., "if you see Robot 17 use a widget of a certain type then do the same, but if you see Robot 5 do it then never mind") is similarly motivated. In a dynamic society agents appear and disappear, denying the designer the ability to anticipate membership in advance. One can sometimes refer to the *roles* of agents (such as Head Robot), and have them treated in a special manner, but we will not discuss this interesting aspect here.

Finally, the notion of "personal history" can be further honed. We will assume that the agent has access to the action he has taken and the reward he received at each instance. One could assume further that the agent observes the choices of others in the games in which he participated, and perhaps also their payoffs. But we will look specifically at an action-selection rule that does not make this assumption. This rule, called the *highest cumulative reward (HCR)* rule, is the following learning procedure:

highest cumulative reward (HCR)

1. Initialize the cumulative reward for each action (e.g., to zero).
2. Pick an initial action.

3. Play according to the current action and update its cumulative reward.
4. Switch to a new action iff the total payoff obtained from that action in the latest m iterations is greater than the payoff obtained from the currently chosen action in the same time period.
5. Go to step 3.

The parameter m in the procedure denotes a finite bound, but the bound may vary. HCR is a simple and natural procedure, but it admits many variants. One can consider rules that use a weighted accumulation of feedback rather than simple accumulation, or ones that normalize the reward somehow rather than looking at absolute numbers. However even this basic rule gives rise to interesting properties. In particular, under certain conditions it guarantees convergence to a "good" convention.

Theorem 7.7.16 *Let g be a symmetric game as defined earlier, with $x > 0$ or $y > 0$ or $x = y > 0$, and either $u < 0$ or $v < 0$ or $x < 0$ or $y < 0$. Then if all agents employ the HCR rule, it is the case that for every $\epsilon > 0$ there exists an integer δ such that after δ iterations of the process the probability that a social convention is reached is greater than $1 - \epsilon$. Once a convention is reached, it is never left. Furthermore, this convention guarantees to the agent a payoff which is no less than the maxmin value of g.*

There are many more questions to ask about the evolution of conventions: How quickly does a convention evolve? How does this time depend on the various parameters, for example m, the history remembered? How does it depend on the initial choices of action? How does the particular convention reached—since there are many—depend on these variables? The discussion below points the reader to further reading on this topic.

7.8 History and references

There are quite a few broad introductions to, and textbooks on, single-agent learning. In contrast, there are few general introductions to the area of *multiagent* learning. Fudenberg and Levine [1998] provide a comprehensive survey of the area from a game-theoretic perspective, as does Young [2004]. A special issue of the *Journal of Artificial Intelligence* [Vohra and Wellman, 2007] looked at the foundations of the area. Parts of this chapter are based on Shoham et al. [2007] from that special issue. Some of the specific references are as follows.

Fictitious play was introduced by Brown [1951] and Robinson [1951]. The convergence results for fictitious play in Theorem 7.2.5 are taken respectively from Robinson [1951], Nachbar [1990], Monderer and Shapley [1996b] and Berger [2005]. The *non*-convergence example appeared in Shapley [1964].

Rational learning was introduced and analyzed by Kalai and Lehrer [1993]. A rich literature followed, but this remains the seminal paper on the topic.

Single-agent reinforcement learning is surveyed in Kaelbling et al. [1996]. Some key publications in the literature include Bellman [1957] on value iteration

in known MDPs, and Watkins [1989] and Watkins and Dayan [1992] on Q-learning in unknown MDPs. The literature on multiagent reinforcement learning begins with Littman [1994]. Some other milestones in this line of research are as follows. Littman and Szepesvari [1996] completed the story regarding zero-sum games, Claus and Boutilier [1998] defined belief-based reinforcement learning and showed experimental results in the case of pure coordination (or team) games, and Hu and Wellman [1998], Bowling and Veloso [2001], and Littman [2001] attempted to generalize the approach to general-sum games. The R-max algorithm was introduced by Brafman and Tennenholtz [2002], and its predecessor, the E3 algorithm, by Kearns and Singh [1998].

The notion of no-regret learning can be traced to Blackwell's approachability theorem [Blackwell, 1956] and Hannan's notion of Universal Consistency [Hannan, 1957]. A good review of the history of this line of thought is provided in Foster and Vohra [1999]. The regret-matching algorithm and the analysis of its convergence to correlated equilibria appears in Hart and Mas-Colell [2000]. Modifications of fictitious play that exhibit no regret are discussed in Fudenberg and Levine [1995] and Fudenberg and Levine [1999].

Targeted learning was introduced in Powers and Shoham [2005b], and further refined and extended in Powers and Shoham [2005a] and Vu et al. [2006]. (However, the term *targeted learning* was invented later to apply to this approach to learning.)

The replicator dynamic is borrowed from biology. While the concept can be traced back at least to Darwin, work that had the most influence on game theory is perhaps Taylor and Jonker [1978]. The specific model of replicator dynamics discussed here appears in Schuster and Sigmund [1982]. The concept of evolutionarily stable strategies (ESSs) again has a long history, but was most explicitly put forward in Maynard Smith and Price [1973]—which also introduced the Hawk–Dove game—and figured prominently a decade later in the seminal Maynard Smith [1982]. Experimental work on learning and the evolution of cooperation appears in Axelrod [1984]. It includes discussion of a celebrated tournament among computer programs that played a finitely repeated Prisoner's Dilemma game and in which the simple Tit-for-Tat strategy emerged victorious. Emergent conventions and the HCR rule were introduced in Shoham and Tennenholtz [1997].