Communication complexity as a lower bound for learning in games

Vincent Conitzer conitzer@cs.cmu.edu
Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213

Tuomas Sandholm sandholm@cs.cmu.edu
Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213

Abstract

A fast-growing body of research in the AI and machine learning communities addresses learning in *games*, where there are multiple learners with different interests. This research adds to more established research on learning in games conducted in economics. In part because of a clash of fields, there are widely varying requirements on learning algorithms in this domain. The goal of this paper is to demonstrate how *communication complexity* can be used as a lower bound on the required learning time or cost. Because this lower bound does not assume any requirements on the learning algorithm, it is universal, applying under any set of requirements on the learning algorithm.

We characterize exactly the communication complexity of various solution concepts from game theory, namely Nash equilibrium, iterated dominant strategies (both strict and weak), and backwards induction. This gives the tighest lower bounds on learning in games that can be obtained with this method.

1. Background

The study of multiagent systems in AI is increasingly concerned with settings where the agents are self-interested. Because in such settings, one agent's optimal action depends on the actions other agents take, there is no longer a straightforward notion of when the agent is acting optimally. *Game theory* is concerned with the study of such settings (or *games*). It provides formalizations of games, such as a game in *normal* (or *matrix*) form (where players choose actions simultaneously), or a game in *extensive* form (where players may choose actions sequentially). It also provides *solution concepts*, which, given a game, specify which outcomes are sensible. Examples for normal form games are *Nash equilibrium* (every player should be playing optimally given what the other players are playing); *dominance* (when one strategy always performs better than another, the latter should not be played); and *iterated* dominance (where dominated strategies are sequentially deleted from the game). Examples for extensive form games include all solution concepts for normal form games, and others such as *backwards induction* (solving the game by working backwards from the last actions in the game). Especially the complexity of constructing Nash equilibria has recently received a lot of attention in the AI and CS theory communities (for example, (Kearns et al., 2001; Papadimitriou, 2001; Leyton-Brown & Tennenholtz, 2003; Blum et al., 2003; Littman & Stone, 2003)); some computational research has also been done on iterated dominance (Gilboa et al., 1993).

It is not always possible for the players to immediately play according to a solution concept: often the players need to *learn* how to play, and will only eventually converge to a solution. There are various reasons why learning may be necessary: the players may not know all of the payoffs (or other variables) in the game; the players may not be sophisticated enough to compute the solution concept; or multiple outcomes may be consistent with the solution concept, and the players need to coordinate. (In this paper, we will focus on the most common variant of the first case, where each player only knows her own payoffs.) Often, constraints are imposed on the learning algorithm. One type of constraint is that of *rationality*: a player should try to maximize her own payoffs. The rationality constraint (when present) takes different (nonequivalent) forms, such as requiring convergence to a best response when playing against a stationary player (Singh et al., 2000; Bowling & Veloso, 2002; Conitzer & Sandholm, 2003), or regret minimization in the limit (Hannan, 1957; Freund & Schapire, 1999). Sometimes the rationality constraint is "soft", in the sense that algo-

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

rithms that recommend actions that seem to be in the player's best interest are preferred. Another type of constraint that is always soft is that of *simplicity*. Of course simpler algorithms are always preferred, but this is especially so in learning in games; and to economists, who are trying to model human behavior, in particular.[1] Computer scientists usually honor this constraint, but there are some exceptions (Conitzer & Sandholm, 2003). The reinforcement learning approach to learning in games (Tan, 1993; Littman, 1994; Hu & Wellman, 1998; Singh et al., 2000; Bowling & Veloso, 2002; Wang & Sandholm, 2002; Greenwald & Hall, 2003; Stimpson & Goodrich, 2003) arguably satisfies both properties of rationality and simplicity.

2. Our approach

In this paper, we study the following question. Two players are playing some game (in either normal (matrix) or extensive form). Each player knows its own payoff for every outcome of the game, but not the opponent's. The players are considering some solution concept (such as Nash equilibrium). How much do the players need to communicate to find a solution (if it exists)?

In general, the players may not want to communicate their payoffs truthfully, because this may cause the other player to choose an action in the game that is disadvantageous to the communicating player. If this is the case, a successful communication may not even be feasible. In this paper, we assume that the players are *completely cooperative* in their communication. That is, they are solely concerned with computing a solution and take no heed of whether this solution is advantageous to them.

Moreover, it may be the case that communication is restricted. For example, it may be the case that communication can only take place through playing the game, by observing the other player's actions (as is usually the case in learning in games). It is possible to implement any communication protocol in this model, by setting up an encoding scheme, encoding bits to be communicated as actions taken in the game. Nevertheless, the players may be reluctant to follow such a protocol, because it may force them to play actions in the game that are highly disadvantageous to themselves while the learning is taking place. In this paper, we do not directly address this, and simply look for the lowest communication complexity in terms of bits.

In spite of these simplifications, there are at least two reasons why the question studied in this paper is important. The first is straightforward:

1. The communication may in fact not be between the actual players of the game, but rather between two other parties (each of which is an expert on one of the players in the game). If the only interest of these parties is to predict the outcome of the game (according to the solution concept under study), so that they will be completely cooperative; and the cost of their communication is the number of bits sent; then our model is accurate.

The second reason is (in our opinion) more interesting:

2. The communication necessary to compute the solution is a *lower bound* on the communication that takes place in *any* learning algorithm that is guaranteed to converge to this solution concept. (This is assuming that in the learning algorithm, the players also do not have access to each other's payoffs.) Combining this with an upper bound on the communication in a single round of the game, we obtain a lower bound on the (worst-case) number of rounds before convergence. For instance, if in an $n \times n$ matrix game, the solution concept requires the communication of $\Omega(f(n))$ bits in the worst case (in our cooperative model), then $\Omega(\frac{f(n)}{\log(n)})$ rounds are required in the worst case to converge to the solution concept. This is because each round, the only communication a player receives from the other player is which action she chose,[2] which is a communication of at most $\log(n)$ bits (or $2\log(n)$ bits counting both players' communication). Given how different the (rationality, simplicity, and other) requirements placed on algorithms in learning in games are, this is arguably the only truly universal lower bound on learning algorithms' worst-case time to convergence.

If the learning cost is not measured in number of rounds (but rather, for instance, in payoffs lost), our technique can sometimes still be used to get a lower bound on the cost. For instance, if there is a minimum cost m incurred in every round before convergence, we get a lower bound on cost of $\Omega(\frac{mf(n)}{\log(n)})$. Also, if certain actions in the game are excessively costly, the learning algorithm should avoid them altogether. Thus, if there are only $n' < n$ reasonable (not excessively costly) actions for each player to take, then the lower bound on the number of rounds increases to $\Omega(\frac{f(n)}{\log(n')})$ (and the previous lower bound on cost increases to $\Omega(\frac{mf(n)}{\log(n')})$).

[1]For an overview of the work in learning in games in economics, see (Fudenberg & Levine, 1998).

[2]In some models of learning in games, not even this is revealed! Of course this can only cause the number of rounds required to go up.

3. Communication complexity

We first review some elementary communication complexity. In this paper, we focus on the two-party model introduced by Yao (Yao, 1979). We follow the presentation in (Kushilevitz & Nisan, 1997).

In Yao's two-party communication model, one party holds input x, and the other holds input y. They seek to compute a binary function $f(x, y)$. The parties alternatingly[3] send bits, according to some protocol. Once the protocol terminates, it should return a value for $f(x, y)$ based on the communicated bits.

Definition 1 *In a* deterministic *protocol, the next bit sent is a function only of the bits sent so far and the sender's input.* $D(f)$ *is the worst-case number of bits sent in the best correct deterministic protocol for computing* f*. In a* nondeterministic *protocol, the communicated bits may additionally depend on nondeterministic choices. For* $z \in \{0, 1\}$*, a nondeterministic protocol for* z *is correct if it always returns* $1 - z$ *when* $f(x, y) = 1 - z$*, and for any* x, y *with* $f(x, y) = z$*, it returns* z *for at least one sequence of nondeterministic choices.* $N^z(f)$ *is the worst-case number of bits sent in the best correct nondeterministic protocol for* z*.*

Because any correct deterministic protocol is also a correct nondeterministic protocol, we have for any function f, and for any z, that $D(f) \geq N^z(f)$. To prove lower bounds on communication complexity, there are numerous techniques (Kushilevitz & Nisan, 1997). However, for the purposes of this paper, we will only need one: that of a *fooling set*. This technique actually proves lower bounds even on nondeterministic communication complexity (and thus also on randomized communication complexity).

Definition 2 *A* fooling set *for value* z *is a set of input pairs* $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ *such that for any* i*,* $f(x_i, y_i) = z$*, but for any* $i \neq j$*, either* $f(x_i, y_j) \neq z$ *or* $f(x_j, y_i) \neq z$*.*

Theorem 1 (Known) *If* z *has a fooling set of size* m*, then* $N^z(f) \geq \log(m)$*.*

4. Normal form games

For the larger part of this paper, we will be concerned with arguably the simplest formalization of games: *normal form* (also known as *matrix*) games.

Definition 3 *In a 2-player* $n \times n$ normal form game*, each player* i *has a set of* (pure) strategies $N = \{1, 2, \ldots, n\}$*, and a* utility (or payoff) function $u_i : N \times N \to \mathbb{R}$*. (We will refer to the players as row (*r*) and column (*c*).)*

[3]The requirement that the bits must be alternatingly sent increases the communication by at most a factor 2.

In the problems below, the players seek to compute a binary function of the game (for instance, whether it has a pure-strategy Nash equilibrium). Each player knows only her own payoff function. That is, the players' payoff functions correspond to the x and y inputs.

4.1. Nash equilibrium

We first study perhaps the best-known solution concept: Nash equilibrium.

Definition 4 *A 2-player game has a* (pure-strategy) Nash equilibrium *if there exist* $i, j \in N$ *such that for any* i'*,* $u_r(i, j) \geq u_r(i', j)$ *and for any* j'*,* $u_c(i, j) \geq u_c(i, j')$*.*

We now define the binary function we seek to compute.

Definition 5 *The function* Na *returns* 1 *if the game has a pure-strategy Nash equilibrium.*

We first give a simple upper bound on the deterministic communication complexity of Na.

Theorem 2 $D(Na)$ *is* $O(n^2)$*.*

Proof: In the protocol, each player communicates, for every entry of the payoff matrix, whether her strategy corresponding to that entry is a best response to the other player's strategy corresponding to that entry—that is, whether she would deviate from that entry. (So, each player communicates one bit per entry.) The game has a pure-strategy equilibrium if and only if for some entry, neither player would deviate from it. ■

We now show a matching lower bound on the nondeterministic communication complexity of Na.

Theorem 3 $N^0(Na)$ *is* $\Omega(n^2)$*, even if all payoffs are either* 0 *or* 1*.*

Proof: We will exhibit a fooling set of size $\Theta(2^{(n^2)})$. Consider the set S of all $n \times n$ matrix games where every entry's payoff vector is either $(0, 1)$ or $(1, 0)$—there are $2^{(n^2)}$ such games. Among these, consider the subset S' that have no row consisting only of $(1, 0)$s and no column consisting only of $(0, 1)$s. $|S'|$ is still $\Theta(2^{(n^2)})$, for the following reason. Suppose we randomly choose a game from S. The probability of any particular row having only $(1, 0)$s (or any particular column having only $(0, 1)$s) is 2^{-n}. It follows that the probability of at least one row having only $(1,0)$s or at least one column having only $(0,1)$s, is at most $2n \cdot 2^{-n}$, which is negligible. So only a negligible fraction of the games in S are not in S'. None of the games in S' have a pure-strategy Nash equilibrium, for the following reason. For any entry in the matrix, one of the players receives 0. If it is the row player, there is another entry in the same column giving her a payoff of 1, and she will want to switch to that entry. If it is the column player, there is another entry in the same row giving her a payoff of 1, and she will want to switch to that entry. Now consider two games $s_1, s_2 \in S'$ with $s_1 \neq s_2$. The games must disagree on at least one entry; say (without loss of generality) that the entry is $(1,0)$ for s_1 and $(0,1)$ for s_2. Then, if we let s_{12} be the game that has the row player's payoffs of s_1 and the column player's payoffs of s_2, the entry under discussion is $(1,1)$ in s_{12}. Because a payoff greater than 1 never occurs in s_{12}, this entry is a pure-strategy Nash equilibrium. ∎

To give an example of how this translates into a bound on learning in games, we can conclude that every multiagent learning algorithm that converges to a pure-strategy Nash equilibrium (if one exists) has a worst-case convergence time of $\Omega(\frac{n^2}{\log(n)})$ rounds (given that the players do not know each other's payoffs).

Communicating the best response function in Theorem 2 is hard because it is set-valued—there can be multiple best responses to a strategy. Next, we investigate what happens if at least one of the players always has a unique best response. Let U be the subset of games where the column player has a unique best response against every (pure) strategy for the row player, and let $Na_{|U}$ be the restriction of Na to such games.

Theorem 4 *$D(Na_{|U})$ is $O(n \log(n))$.*

Proof: In the protocol, the column player communicates her (unique!) best response to each of the row player's pure strategies (a communication of $\log(n)$ per strategy). After this, the row player can determine if a pure strategy Nash equilibrium exists (if and only if for one of the row player's pure strategies i, i is a best response to the column player's best reponse to i), and can communicate this to the column player. ∎

Theorem 5 *$N^0(Na_{|U})$ is $\Omega(n \log(n))$, even if all payoffs are either 0 or 1.*

Proof: We will exhibit a fooling set of size $n!$. This will prove the theorem, because $n \log(n)$ is $\Theta(\log(n!))$. For every permutation $\pi : N \to N$, consider the following game. When the row player plays i and the column player plays j, the row player gets a utility of 1 if $\pi(i) \neq j$, and 0 otherwise; the column player gets a utility of 0 if $\pi(i) \neq j$, and 1 otherwise. Because this is a zero-sum game, and because for each player, against any opponent strategy, there is a strategy that wins against this opponent strategy, there is no pure-strategy equilibrium. All that remains to show is that if we mix the payoffs of two of these games, that is, we define the row player's payoffs according to π_1, and the column player's payoffs according to $\pi_2 \neq \pi_1$, there is a pure-strategy Nash equilibrium. Let i be such that $\pi_1(i) \neq \pi_2(i)$. Then the strategy pair $(i, \pi_2(i))$ gives the row player utility 1 (because $\pi_1(i) \neq \pi_2(i)$), and the column player utility 1 also. Because 1 is the highest utility in the game, this is a Nash equilibrium. ∎

Interestingly, slight adaptations of all the proofs in this subsection also work for *Stackelberg equilibrium*, where the row player moves first. (Here the computational question is defined as "Should the row player play her first action?") Thus, Stackelberg equilibrium has the same communication complexity in each case. We omit the proofs because of space constraint.

4.2. Iterated dominance

We now move on to the notions of *dominance* and *iterated dominance*. The idea is that if one strategy always performs better than another, we may eliminate the latter from consideration.

Definition 6 *In a 2-player game, one strategy is said to* strictly dominate *another strategy for a player if the former gives a strictly higher payoff against any opponent strategy. One strategy is said to* weakly dominate *another strategy if the former gives at least as high a payoff against any opponent strategy, and a strictly higher payoff against at least one opponent strategy.*

Sometimes, *mixed* strategies (probability distributions over pure strategies) are allowed to dominate other strategies. A mixed strategy's payoff against an opponent strategy is simply its *expected* payoff.

It is rarely the case that one strategy dominates all others. However, once we eliminate a strategy from consideration, new dominances may appear. This sequential process of eliminating strategies is known as *iterated dominance*.

Definition 7 *With* iterated dominance*, dominated strategies are sequentially removed from the game. (Here, removing strategies may lead to new dominance* *relations.) A game is said to be* solvable by iterated dominance *if there is a sequence of eliminations that eliminates all but one strategy for each player.*

While the definition of iterated dominance is conceptually the same for both strict and weak dominance, the concept technically differs significantly depending on which form of dominance is used. Iterated strict dominance is known to be *path-independent*: that is, eventually the same strategies will remain regardless of the order in which strategies are eliminated. Iterated weak dominance, on the other hand, is known to be *path-dependent*: which strategies eventually remain depends on the elimination order. In fact, determining whether a game is solvable by iterated weak dominance is NP-complete (Gilboa et al., 1993).

We first study iterated strict dominance.

Definition 8 *The function isd returns* 1 *if the game can be solved using iterated strict dominance.*

Theorem 6 $D(isd)$ *is* $O(n \log(n))$*, whether or not elimination by mixed strategies is allowed.*

Proof: In the protocol, the players alternatingly communicate one of their dominated strategies, which the players then consider removed from the game; or communicate that no such strategy exists. (We observe that this requires the communication of $O(\log(n))$ bits.) The protocol stops once both players successively communicate that they have no dominated strategy. We observe that each player can get at most $2n$ turns in this protocol. Hence the number of bits communicated is $O(n \log(n))$. ■

Theorem 7 $N^1(isd)$ *is* $\Omega(n \log(n))$*, even if all payoffs are either* 0 *or* 1*, whether or not elimination by mixed strategies is allowed.*

Proof: We will exhibit a fooling set of size $n!$. This will prove the theorem, because $n \log(n)$ is $\Theta(\log(n!))$. For every permutation $\pi : N \to N$, consider the following game. The row player's payoff when the row player plays i and the column player plays j is 0 if $\pi(i) \leq j$, and 1 otherwise—unless $\pi(i) = n$, in which case the row player's payoff is always 1. The column player's payoff when the row player plays i and the column player plays j is 0 if $j < \pi(i)$, and 1 otherwise. (We observe that there is a weak dominance relation between any pair of strategies by the same player, and thus allowing for dominance by mixed strategies cannot help us—we may as well take the most weakly dominant strategy in the support. So, we can restrict attention to elimination by pure strategies in the rest of the proof.) Because the row player always gets a payoff of 1 playing $\pi^{-1}(n)$, and the column player always gets a payoff of 1 playing n, we can eliminate any strategy that always gets a player a payoff of 0 against the opponent's remaining strategies. Thus, the row player can eliminate $\pi^{-1}(1)$; then the column player can eliminate 1; then the row player can eliminate $\pi^{-1}(2)$; then the column player can eliminate 2; *etc.*, until all but $\pi^{-1}(n)$ for the row player and n for the column player have been eliminated. So every one of these games can be solved by iterated strict dominance. All that remains to show is that if we mix the payoffs, that is, we define the row player's payoffs according to π_1, and the column player's payoffs according to $\pi_2 \neq \pi_1$, the game is not solvable by iterated strict dominance. Let k be the lowest number such that $\pi_1^{-1}(k) \neq \pi_2^{-1}(k)$ (we observe that $k \leq n-1$). Because iterated strict dominance is path-independent, we may assume that we start eliminating strategies as before for as long as possible. Thus, we will have eliminated strategies $\pi_1^{-1}(1) = \pi_2^{-1}(1), \pi_1^{-1}(2) = \pi_2^{-1}(2), \ldots, \pi_1^{-1}(k-1) = \pi_2^{-1}(k-1), \pi_1^{-1}(k)$ for the row player, and strategies $1, 2, \ldots, k-1$ for the column player. However, at this point, $\pi_2^{-1}(k)$ for the row player will not have been eliminated, so playing k (or any other remaining strategy) will get the column player 1 against $\pi_2^{-1}(k)$, and can thus not be eliminated. Similarly, any remaining strategy will get the row player 1 against k, and can thus not be eliminated. Because $k \leq n-1$, the game cannot be solved by iterated strict dominance. ■

We now move on to iterated weak dominance.

Definition 9 *The function iwd returns* 1 *if the game can be solved using iterated weak dominance.*

We first give an upper bound on the *nondeterministic* communication complexity.

Theorem 8 $N^1(iwd)$ *is* $O(n \log(n))$*, whether or not elimination by mixed strategies is allowed.*

Proof: As in Theorem 6, the players alternatingly communicate an eliminated strategy (they nondeterministically choose one from the weakly dominated strategies at that point), and return 1 if they reach a solution. Because iterated weak dominance is path-dependent, whether a solution is reached depends on the nondeterministic choices made; but if the game is solvable, then at least for some sequence of nondeterministic choices, they will reach a solution. ■

Assuming P≠NP, any *deterministic* communication protocol for determining whether a game is solvable by iterated weak dominance must either have an exponential communication complexity, or require exponential computation per communication step by the players. (Because otherwise, we would have a polynomial-time algorithm for determining whether a game is solvable by iterated weak dominance, which is NP-complete (Gilboa et al., 1993).) We can avoid this by restricting attention to the following subset of games, where path-dependence is partially assumed away. Let I be the subset of games where either no solution by iterated weak dominance exists, or any elimination path will lead to a solution; and let $iwd_{|I}$ be the restriction of iwd to such games.

Theorem 9 *$D(iwd_{|I})$ is $O(n\log(n))$, whether or not elimination by mixed strategies is allowed.*

Proof: Because (by assumption) any elimination path will do, the approach in Theorem 6 is applicable. ∎

The following theorem shows that this is the best we can do even with the restriction to the set I (and also that the nondeterministic algorithm given before is the best we can do).

Theorem 10 *$N^1(iwd_{|I})$ is $\Omega(n\log(n))$, even if all payoffs are either 0 or 1, whether or not elimination by mixed strategies is allowed.*

Proof: We first observe that when all the payoffs are 0 or 1, allowing for weak dominance by mixed strategies does not allow us to perform any more eliminations. (If mixed strategy σ weakly dominates pure strategy σ', then all the pure strategies in the support of σ must get a payoff of 1 against any strategy that σ' gets a payoff of 1 against. Moreover, at least one pure strategy in the support must receive a strictly better payoff than σ' against at least one opponent strategy. But then this pure strategy also weakly dominates σ'.) Thus, we can restrict attention to elimination by pure strategies in the rest of this proof.

For even n ($n = 2l$), we will exhibit a fooling set of size $(\frac{n}{2}-1)!$. This will prove the theorem, because $n\log(n)$ is $\Theta(\log((\frac{n}{2}-1)!))$. (Note that $\frac{n}{2}\log(\frac{n}{2})$ is $\Theta(n\log(n))$.) For every permutation $\pi : L \rightarrow L$ with $\pi(l) = l$, consider the following game. When the row player plays i and the column player plays j, both players receive 0 when $j > l$ (half of the row player's strategies are dummy strategies). Otherwise, the row player receives a payoff of 1 whenever $j \in \{\pi(i), \pi(i)-1, l+\pi(i)-1, \pi(i)-2\}$, and 0 otherwise. The column player receives a payoff of 1 whenever $j \in \{\pi(i), l+\pi(i)\}$, and 0 otherwise—unless $j = n = 2l$, in which case the column player always receives 0 (another dummy strategy).

In each of these games, both players can first eliminate the dummy strategies; then, the row player can eliminate $\pi^{-1}(1)$ using $\pi^{-1}(2)$; then the column player can eliminate 1 and $l{+}1$ using (for example) l; then the row player can eliminate $\pi^{-1}(2)$ using $\pi^{-1}(3)$; then the column player can eliminate 2 and $l{+}2$ using l; *etc.*, until only $\pi^{-1}(l) = l$ is left for the row player, and only l is left for the column player. So every one of these games can be solved by iterated weak dominance. Moreover, any sequence of eliminations will arrive at this solution, for the following reasons. $\pi^{-1}(l)$ $(= l)$ for the row player and l for the column player are the unique best responses to each other and hence cannot be eliminated. Thus, eventually all the dummy strategies must be deleted. For the same reason, whenever $\pi^{-1}(t)$ has been deleted, $l+t$ and t must eventually be deleted. Furthermore, $l+t$ must survive for the column player as long as $\pi^{-1}(t)$ survives for the row player, because $l+t$ and t are the only best responses to $\pi^{-1}(t)$, and neither can dominate the other. Finally, for the row player, $\pi^{-1}(t+1)$ cannot be eliminated before $\pi^{-1}(t)$, because as long as $\pi^{-1}(t)$ survives, $l+t$ must survive for the column player, and $\pi^{-1}(t+1)$ is the unique best response to $l+t$. Thus, for the smallest $t < l$ such that $\pi^{-1}(t)$ survives, eventually $l+t-1$, $t-1$, and $t-2$ must be eliminated for the column player, and then eventually $\pi^{-1}(t+1)$ must eliminate $\pi^{-1}(t)$ (because it performs better against $l{+}t$, which cannot yet have been eliminated). So any elimination path will lead to the solution.

All that remains to show is that if we mix the payoffs, that is, we define the row player's payoffs according to π_1 and the column player's payoffs according to $\pi_2 \neq \pi_1$, the game is not solvable by iterated weak dominance. Suppose there is a solution by iterated weak dominance. Because the strategies labeled l are unique best responses against each other (regardless of the permutations), neither can ever be eliminated, so they must constitute the solution.

We first claim that the row player's non-dummy strategies, and the column player's non-dummy strategies of the form $l+k$, must be alternatingly eliminated. That is, two non-dummy row player strategies cannot be eliminated without a non-dummy column player strategy of the form $l+k$ being eliminated somewhere inbetween, and two non-dummy column player strategies of the form $l+k$ cannot be eliminated without a non-dummy row player strategy being eliminated somewhere inbetween. Moreover, each non-dummy row player strategy that is eliminated must be the best response to the last non-dummy column player strategy of the form $l+k$ that was eliminated (with the exception of $\pi_1^{-1}(1)$); and vice versa, each non-dummy column player strategy of the form $l+k$ that is eliminated must be the best response to the last non-dummy row player strategy that was eliminated. This is so because each non-dummy row player strategy (besides $\pi_1^{-1}(1)$) is the unique best response against some non-dummy column player strategy of the form $l+k$; and each non-dummy column player strategy of the form $l+k$ is the almost unique best response against some non-dummy row player strategy. ("Almost" because k is also a best response, but k can never eliminate $l+k$ because it always performs identically.) Thus the only way of eliminating these strategies is to first eliminate $\pi_1^{-1}(1)$ for the row player, then the best response to that strategy among the column player strategies of the form $l+k$, then the best response to *that* strategy, *etc.* (Other strategies are eliminated inbetween this.)

Now, we claim that in the solution, the non-dummy column player strategies of the form $l+k$ must be eliminated in the order $l+1, l+2, \ldots, 2l-1$. For suppose not: then let $l+k$ be the first eliminated strategy such that $l+k-1 \geq l+1$ has not yet been eliminated. (We note that $k \geq 2$.) By the above, the next non-dummy row player strategy to be eliminated should be $\pi_1^{-1}(k+1)$. However, $l+k-1$ and $l+k+1$ have not yet been eliminated,[4] and thus, $k-1$ and $k+1$ cannot be eliminated before $\pi_1^{-1}(k+1)$ (because if $k-1$ (or $k+1$) could be eliminated, then $l+k-1$ (or $l+k+1$) could also be eliminated at this point, contradicting the alternation in the elimination proven above). But $\pi_1^{-1}(k+1)$ is the only strategy that gets the row player a utility of 1 against both of $k-1$ and $k+1$, so it cannot be eliminated. Thus the elimination cannot proceed. This proves the claim. But if this is the elimination order, it follows that $1 = \pi_2(\pi_1^{-1}(1))$ (because $l+1$ is the best reponse against $\pi_1^{-1}(1)$), $2 = \pi_2(\pi_1^{-1}(2))$ (because $l+2$ is the best response against $\pi_1^{-1}(2)$, which is the best response against 1), *etc.* Thus the permutations must be the same, as was to be shown. ■

5. Extensive form games

For the (short) remainder of this paper, we will focus on a different formalization of games: *extensive form* games—games that can be represented in tree form.

Definition 10 *A 2-player* (full information) extensive form game *is given by a tree with n nodes (one of which is the root), a specification of which player moves at each node, and a payoff from* $\mathbb{R}$ *for each player at every leaf.*

[4] If $l+k+1 = 2l$, it may have been eliminated, but $k+1$ will still be there in this case, too.

5.1. Backwards induction

The simplest solution concept for games in extensive form is that of *backwards induction*, where the best action to take is determined at every node, starting from the bottom nodes and working upwards. To make the "best" action uniquely defined, we will restrict attention to the subset of extensive form games E in which no player gets the same payoff at two different leaves.

Definition 11 *In the backwards induction solution, each node is labeled with one of its children, indicating which action the player at this node should take; under the constraint that for each player, each of her actions should give her the maximal payoff given the actions specified lower in the tree.*

Definition 12 *The function $b_{|E}$ returns 1 if in the backwards induction solution, player 1 chooses her leftmost action at the root.*

Theorem 11 *$D(b_{|E})$ is $O(n)$.*

Proof: For each choice node in the tree that is not followed by another choice node (that is, the bottom choice nodes), the corresponding player communicates which action she would take at this choice node. As a result, now both players know their valuations for the bottom choice nodes. Thus, in the next stage, for each choice node in the tree that is followed by at most one more choice node (that is, the "second-to-bottom" choice nodes), the corresponding player can communicate which action she would take here. We can continue this process until the players know which action player 1 takes at the root. The communication can be achieved by labeling each edge in the tree as either 0 (for not taken) or 1 (for taken), in the bottom-to-top order just described. ■

Theorem 12 *$N^1(b_{|E})$ is $\Omega(n)$. (Even when the tree has depth 2.)*

Proof: Omitted because of space constraint. ■

6. Conclusions and future research

In learning in games, there are widely varying requirements on learning algorithms. We demonstrated how *communication complexity* can be used as a lower bound on the required learning time or cost. Because this lower bound does not assume any requirements on the learning algorithm, it is universal, applying under any set of requirements on the learning algorithm.

We characterized exactly the communication complexity of various solution concepts from game theory, giving the tighest lower bounds on learning these concepts that can be obtained with this method. We showed that the communication complexity of finding a pure-strategy Nash equilibrium in an $n \times n$ game is $\Theta(n^2)$ (but only $\Theta(n \log(n))$ when one of the players always has a unique best response to any strategy); the communication complexity of iterated strict dominance is $\Theta(n \log(n))$ (whether or not dominance by mixed strategies is allowed); the communication complexity of iterated weak dominance (for games in which solvability is path-independent) is $\Theta(n \log(n))$ (whether or not dominance by mixed strategies is allowed); and the communication complexity of backwards induction in a tree with n nodes is $\Theta(n)$. (Interestingly, the size of the payoffs is not a factor in any of these complexities.) In each case, we showed the lower bound holds even for nondeterministic communication, and gave a deterministic protocol that achieved the bound.

There are various directions for future research. Can the lower bounds presented in this paper be achieved by learning algorithms with additional desirable properties? (The most important such property would be some measure of *rationality*: the player should attempt to do reasonably well in the game even when still learning. Also, we may wish to add the following constraint: a player should not be able to make the solution that the players eventually converge to more advantageous to herself, by not following the learning algorithm.) If this is not possible, what are *minimal* restrictions on the learning algorithm that will allow us to strengthen our lower bounds and close the gap? (For instance, is there a weak rationality criterion that all sensible notions of rationality should satisfy, and that strengthens the lower bound significantly?) This would constitute a new branch of communication complexity theory, where communication is nontrivially constrained.

Another interesting direction for future research is to investigate whether learning algorithms can do better than the bounds presented in this paper on specific distributions of games (perhaps drawn from the real world). After all, the lower bounds presented in this paper are worst-case results. We also did not study the communication complexity of computing a mixed-strategy Nash equilibrium. (We do observe that because a mixed-strategy Nash equilibrium always exists, the existence question is trivial in this case.) Yet another possibility is to study solution concepts such as Nash equilibrium for the repeated game rather than for the one-shot game. Finally, one can study whether and how things change if we impose computational constraints on the agents.

7. Acknowledgements

This material is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678.

References

Blum, B., Shelton, C. R., & Koller, D. (2003). A continuation method for Nash equilibria in structured games. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, Mexico.

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

Conitzer, V., & Sandholm, T. (2003). AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *International Conference on Machine Learning* (pp. 83–90).

Freund, Y., & Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games & Econ. Behavior*, *29*, 79–103.

Fudenberg, D., & Levine, D. (1998). *The theory of learning in games*. MIT Press.

Gilboa, I., Kalai, E., & Zemel, E. (1993). The complexity of eliminating dominated strategies. *Mathematics of Operation Research*, *18*, 553–565.

Greenwald, A., & Hall, K. (2003). Correlated Q-learning. *International Conference on Machine Learning* (pp. 242–249).

Hannan, J. (1957). Approximation to Bayes risk in repeated play. vol. III of *Contributions to the Theory of Games*, 97–139.

Hu, J., & Wellman, M. P. (1998). Multiagent reinforcement learning: Theoretical framework and an algorithm. *International Conference on Machine Learning* (pp. 242–250).

Kearns, M., Littman, M., & Singh, S. (2001). Graphical models for game theory. *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Kushilevitz, E., & Nisan, N. (1997). *Communication complexity*. Cambridge University Press.

Leyton-Brown, K., & Tennenholtz, M. (2003). Local-effect games. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, Mexico.

Littman, M., & Stone, P. (2003). A polynomial-time Nash equilibrium algorithm for repeated games. *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)* (pp. 48–54).

Littman, M. L. (1994). Markov games as a framework for multiagent reinforcement learning. *International Conference on Machine Learning* (pp. 157–163).

Papadimitriou, C. (2001). Algorithms, games and the Internet. *STOC* (pp. 749–753).

Singh, S., Kearns, M., & Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. *Conference on Uncertainty in Artificial Intelligence (UAI)* (pp. 541–548).

Stimpson, J., & Goodrich, M. (2003). Learning to cooperate in a social dilemma: A satisficing approach to bargaining. *International Conference on Machine Learning* (pp. 728–735).

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. *International Conference on Machine Learning* (pp. 330–337).

Wang, X., & Sandholm, T. (2002). Reinforcement learning to play an optimal Nash equilibrium in team Markov games. *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. Vancouver, Canada.

Yao, A. C. (1979). Some complexity questions related to distributed computing. *Proceedings of the 11th ACM symposium on theory of computing (STOC)* (pp. 209–213).