Artificial Intelligence 280 (2020) 103219

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

When autonomous agents model other agents: An appeal for altered judgment coupled with mouths, ears, and a little more tape

Jacob W. Crandall

Computer Science Department, Brigham Young University, Provo, UT 84602, United States of America

ARTICLE INFO

Article history:
Received 12 February 2019
Received in revised form 30 August 2019
Accepted 2 December 2019
Available online 16 December 2019

Keywords:
Agent modeling
Autonomous agents
Multi-agent systems

ABSTRACT

Agent modeling has rightfully garnered much attention in the design and study of autonomous agents that interact with other agents. However, despite substantial progress to date, existing agent-modeling methods too often (a) have unrealistic computational requirements and data needs; (b) fail to properly generalize across environments, tasks, and associates; and (c) guide behavior toward inefficient (myopic) solutions. Can these challenges be overcome? Or are they just inherent to a very complex problem? In this reflection, I argue that some of these challenges may be reduced by, first, modeling alternative processes than what is often modeled by existing algorithms and, second, considering more deeply the role of non-binding communication signals. Additionally, I believe that progress in developing autonomous agents that effectively interact with other agents will be enhanced as we develop and utilize a more comprehensive set of measurement tools and benchmarks. I believe that further development of these areas is critical to creating autonomous agents that effectively model and interact with other agents.

1. Introduction

In her book *Team of Rivals* [1], Doris Kearns Goodwin repeatedly observed that Abraham Lincoln, the sixteenth president of the United States, seemed to have an uncanny ability to know how individuals and groups would react in various circumstances. Lincoln used this talent to benefit both himself and society as a whole. Among other things, he used his ability to understand people to navigate a very tricky political environment in order to bring about the emancipation of slavery in the United States in the 1860s. We imagine a similar vision for AI systems. Autonomous agents can bring about much good if they can effectively interact with people and other autonomous agents. As with Abraham Lincoln, appropriately and accurately understanding and modeling other agents is a necessary capability for realizing this vision.

In their well-written and comprehensive survey [2], Albrecht and Stone reviewed a vast amount of research on autonomous agents modeling other agents. Two things jump out to me from this survey. First, prior work has, over the last several decades, resulted in many elegant algorithms for modeling the behavior of other agents. Second, the collective body of work highlights strengths and weaknesses of many modeling mechanisms, including policy reconstruction, type-based reasoning, agent classification, plan recognition, recursive reasoning, graphical models, group modeling, and other implicit

E-mail address: crandall@cs.byu.edu.

https://doi.org/10.1016/j.artint.2019.103219

modeling mechanisms. Albrecht and Stone noted that no one method appears to dominate all the others. In fact, each category of modeling mechanism has one or more of the following common weaknesses:

- The agent-modeling method is computationally complex, such that it cannot easily scale to real-world problems.
- The agent-modeling method requires more data than can realistically be acquired. As such, constructed models tend to be either incomplete or inaccurate, making it difficult or risky to utilize them in decision-making.
- The agent-modeling method is tedious and unrealistic to construct for arbitrary environments. It is difficult to update when more information becomes available.
- The accuracy of the agent-modeling method is subject to modeling assumptions that are difficult to get right for arbitrary associates[1] and environments. Furthermore, some modeling mechanisms assume that the agents they model are less complex than they are themselves, making it difficult for an autonomous agent to, for example, model an agent that uses the same algorithm as it does.

These common weaknesses demonstrate the difficulty of developing agent-modeling mechanisms that simultaneously (a) produce sufficiently correct models, (b) are computationally feasible, and (c) are sufficiently general to be used effectively by autonomous agents. It is, perhaps, for this reason that agent modeling and other skills of social interaction remain compelling AI research agendas despite the large amount of attention these subjects have already attracted.

Are these challenges insurmountable, unlikely to be solved? Or can we develop general mechanisms that will lead to agents that, in arbitrary situations, adequately model their associates? Are current research directions leading us in the right direction, or are alternative perspectives necessary to address these issues broadly?

While only time is likely to answer these and other related questions, I argue for greater attention to three research thrusts which will, I hope, help mitigate at least some of the previously stated difficulties in agent modeling. I first argue that many efforts have, perhaps, misidentified which agent models are most important. Making different kinds of judgments than we are often prone to make can, I believe, lead autonomous agents to more effectively interact with their associates. Second, I believe that non-binding communication signals are not being given sufficient attention in many scenarios and algorithms considered by AI researchers. While there are a variety of good reasons for this, the absence of these signals in our systems makes accurately modeling other agents in arbitrary environments unrealistic. As such, I maintain that research into developing algorithms that better utilize non-binding communication signals should be more abundant. Finally, I argue that progress in this field will be enhanced as we develop an improved and more diverse set of measurement tools and benchmarks.

2. To model or to be modeled

> *[T]oo much complexity can appear to be total chaos.*
>
> [Robert Axelrod, *The Evolution of Cooperation*]

> *Judge not, that ye be not judged.... [F]irst take the log out of your own eye, and then you will see clearly to take the speck out of your brother's eye.*
>
> [English Standard Version, *Matthew 7:1,5*]

While Abraham Lincoln appears to have possessed an uncanny ability to predict the behavior of others, it has also been argued that Lincoln simultaneously cultivated (whether intentionally or unintentionally) his image. In summarizing remarks by various historians, Adam Grant in *Give and Take* [3] stated: “Lincoln is seen as one of the least self-centered, egotistical, boastful presidents ever.” This combination of attributes (the ability to model and the ability to manage the models other agents had of himself) appear to have together been critical to his ability to bring about change.

Likewise, as it establishes relationships with other agents, an autonomous agent should consider three interrelated subtasks simultaneously:

- *Subtask 1*: Model the other agents in the environment.
- *Subtask 2*: Manage the models that other agents in the environment form about itself (i.e., be modeled).
- *Subtask 3*: Select actions (often with the goal of maximizing the agent's expected utility) given the models created in performing Subtasks 1 and 2.

These subtasks are difficult to separate. Furthermore, attending to or exploiting one of these subtasks often comes at the expense of neglecting one or more of the other subtasks. In fact, prior work (e.g., [4–6]) demonstrates that it is sometimes impossible to perform all three subtasks optimally. Limited computational resources, knowledge, and opportunity dictate that an agent must prioritize one or more of these subtasks.

[1] Throughout this paper, I use the term *associate* to refer to another agent (living or artificial) with whom one interacts. The term is not intended to imply whether or not cooperation with this other agent is desirable.

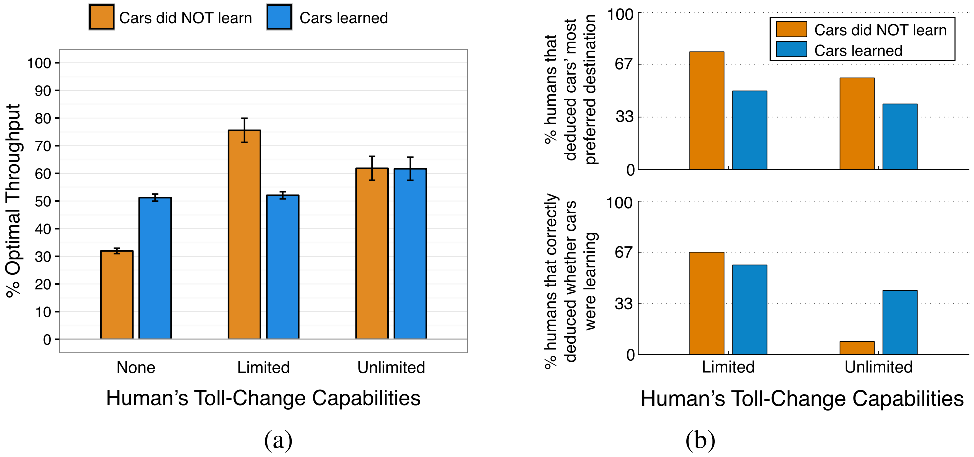


Fig. 1. Results from a study in which people regulated the behavior of driverless cars. (a) The average system throughput of communities of driverless cars regulated by tolls set by people (the regulators) that employed either static or (reinforcement) learning algorithms. (b) The percentage of human regulators that correctly identified characteristics (preferred destination and learning behavior) of the driverless cars. Figures adapted from Shen et al. [13].

Which subtask or subtasks should an autonomous agent that interacts with other agents prioritize? While a clear answer is perhaps unlikely at this point, prior studies provide useful evidence.

2.1. Evidence from prior studies

To give insight into which subtask or subtasks an autonomous agent should prioritize, I review and reflect on two studies conducted in different domains. The first study is Axelrods's decades-old comparison of algorithms in the iterated prisoner's dilemma [7], which remains relevant and fascinating today. In this study, Axelrod invited researchers to submit algorithms to compete in an iterated prisoner's dilemma tournament. Despite many elegant and complex solutions, tit-for-tat (TFT), the simplest algorithm submitted in terms of lines of code, was the highest performing algorithm in multiple iterations of the tournament. TFT's robustness in prisoner's dilemmas has led to broad use and scrutiny. Generalized implementations of the algorithm have been used extensively (e.g., [8–11]), and recent and ongoing work continues to consider how complex machine-learning algorithms can be designed to derive TFT-like behavior (e.g., [12]).

Axelrod attributed TFT's success to several properties: (1) it was never the first to defect, (2) it reciprocated both cooperation and defection in a way that balanced vengeance and forgiveness effectively, and (3) it clearly communicated its strategy to its associates via its actions (i.e., it was easy for its partner to model). Many of the less successful algorithms tried to exploit explicit or implicit models of their associates. Axelrod concluded that these more complex algorithms displayed behaviors that were not easily modeled by their associates, and hence they did not perform as well. On the surface, these conclusions paint a perspective that an autonomous agent should prioritize Subtask 2, and that algorithms focused on exploiting models of their associates (Subtask 1; seemingly eschewing Subtask 2) were often too complex for their own good.

A second study, results of which are summarized in Fig. 1, paints a similar picture. In this study, Shen et al. [13] considered how well people could use tolls to regulate autonomous (driverless) cars in a simple transportation network. Participants in the study (taking the role of regulators of the system) set tolls on roads in the network with the goal of alleviating congestion and maximizing throughput. To investigate which factors resulted in more efficient transportation networks, two societies of autonomous cars were considered. In the first society, each car learned which roads to traverse from its own past experiences using reinforcement learning. In the second society, the cars did not learn from their experiences, but maintained a static model of the world throughout the scenario (other than observing toll changes). Shen et al. also varied the power given to the regulators. Some regulators were allowed to change tolls as often as they desired (unlimited), while other regulators were only allowed a fixed number of toll changes throughout the scenario.

While societies in which the cars learned had the highest throughput in the absence of human regulations, Shen et al. observed that the most effective system was the one in which the cars did not learn and the humans had limited toll-changing capabilities (Fig. 1a). Further analysis indicated that both the adaptive ability of the cars and the human's own complex behavior were associated with human regulators being less likely to accurately model the behavior of the autonomous cars (Fig. 1b).

In many senses, the results of this second study mirror the results of Axelrod's study. By acting on a model of their environment (which included an implicit model of the other agents), the cars using reinforcement learning became too complex for the human regulators to model. As a result, the society of cars that used simple, easily modeled (even if seemingly inferior) routing algorithms were better off.

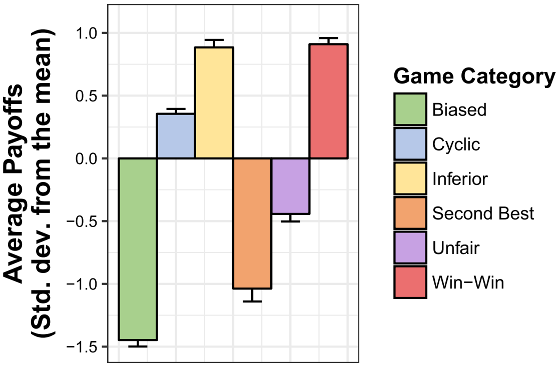


Fig. 2. The relative normalized average payoffs of generalized TFT (gTFT) compared to those of other algorithms when paired with 25 different algorithms in games drawn from six different game categories classified by Bruns [18]. Payoffs are normalized in each game category with respect to the standard deviation from the mean payoff obtained by all players (across all pairings). While gTFT was a high performing algorithm in *Win-Win*, *Inferior* (the category that contains the prisoner's dilemma), and (to a lesser extent) *Cyclic* games, it performed poorly in games drawn from the other three game categories. Figure adapted from Crandall et al. [17] (Supplementary Note 3). (Figure is best viewed in color, which is available in the web version of this article.)

On the surface, these prior results seem to suggest that successful autonomous agents can focus on Subtask 2 with little regard for modeling their associates. Further scrutiny modifies the picture somewhat, however. For example, in the decades following Axelrod's study, much research has continued to study TFT (and generalized variants) and other kinds of algorithms in repeated games. Some of this research has highlighted TFT's deficiencies, such as demonstrating how TFT fails when other agents are not sophisticated enough to model its behavior [14] and identifying strategies that perform better in prisoner's dilemmas under certain circumstances (e.g., [15,14,16]). Furthermore, while very successful in prisoner's dilemmas, generalized TFT is not always as successful in scenarios with different payoff structures and equilibrium characteristics (Fig. 2; [17]). This suggests that the strategic principles identified by Axelrod are not sufficiently general, and may need refinement.

2.2. Alternative perspective: questions dictate agent models and success

While this view of prior results may suggest that autonomous agents may benefit from not making judgments (i.e., models) about their associates, I believe this simplified conclusion is flawed. I argue that autonomous agents simply should make different judgments about the agents they interact with, and that these alternative judgments align better with simultaneously managing other agents' models of one's self.

The judgments an autonomous agent makes about others is dictated by the questions its algorithmic processes seek to answer. These questions thus dictate the agent's behavior and, ultimately, its success. For example, common views of so-called rationality (maximizing expected utility given beliefs) state that an autonomous agent should seek to learn to play *a strategy that maximizes its expected utility given the strategy played by its associate*. Algorithms that pursue this agenda often (either explicitly or implicitly) place priority on answering the following two questions:

Q1: *What is my associate's strategy?*
Q2: *Am I playing the best strategy given my model of my associate's strategy?*

Algorithms that pursue answers to questions Q1 and Q2 have been dubbed *Followers* [14]. A common characteristic of these algorithms is that they perform effectively when they form sufficiently farsighted and accurate models, but they perform quite poorly when their models are either myopic or incorrect. Thus, since correctly modeling the strategies of non-adaptive or memory-bounded associates is often quite doable, Followers do quite well in such situations (even deriving payoff maximizing strategies). However, as has often been argued, creating good models of adaptive, potentially like-minded, associates is very difficult, in part because their associate's strategies are adapting to their own. In such situations, these algorithms sometimes produce myopic behavior that decidedly does not maximize payoffs.

But questions Q1 and Q2 are not the only set of questions a successful agent can pursue. An alternative way for an autonomous agent to approach an interaction is to "begin with the end in mind" [19]. This approach focuses on finding a "win-win" solution by pursuing an answer to the following question:

Q3: *What desirable[2] outcome is likely to be acceptable to my associate?*

[2] Question Q3 does not directly pursue a clear definition of optimality (as is commonly done in answering question Q2) since there is a difficult trade-off between the *likeliness* that the strategy will lead to a desirable outcome (which can be difficult to quantify) and the actual *quality* of the outcome.

Once an answer to this question is determined, the autonomous agent plays a strategy that promotes the computed outcome. Often, this requires the agent to clearly demonstrate via its actions that it intends to comply with the selected outcome, perhaps on condition that their associate will also do so (Subtask 2). In this way, autonomous agents can, like Lincoln, simultaneously pursue Subtasks 1 and 2.

Leader algorithms [14] (such as TFT and Bully) represent one genre of algorithm that pursues question Q3. Once these algorithms answer this question, they conform with the selected outcome as long as their associate does likewise, and otherwise punish deviations from the outcome to try to promote its value to the associate.

As is seen in the results of Axelrod's prisoner's dilemma tournaments, strategies that pursue question Q3 *can* be incredibly effective and simple. On the other hand, such algorithms can also be quite ineffective (see, for example, Fig. 2). Leader algorithms fail in part because they do not internalize that their associate will not, for one reason or another, comply with the outcome they selected.

To address this weakness, a third set of algorithms, which I call *Builders*, pursue a second question after answering question Q3:

Q4: *Does my associate agree to the outcome I am proposing?*

If the answer is *yes*, Builders continue to act according to the proposed solution. If not, Builders either re-evaluate question Q3 or, in the case of decided failure to build a consensus, resort to a process that focuses on questions Q1 and Q2. In essence, rather than continually pursuing an outcome that their associate is not likely to accept, Builders seek consensus by iteratively pursuing questions Q3 and Q4 until a desirable outcome is accepted by its associate.

Several example Builder algorithms have been described in the literature on repeated games. MetaStrategy, for example, which was proposed by Powers and Shoham [9], begins an interaction by playing generalized TFT, its initial answer to question Q3. After trying this strategy for a fix number of rounds, it evaluates whether or not its payoffs meet or exceed what it would receive if its associate agreed to the solution it proposed (thus, this is an implicit answer to question Q4). If its average payoffs meet or exceed this threshold, it continues to play generalized TFT. Otherwise, it switches to playing Fictitious Play (which pursues answers to questions Q1 and Q2). Powers and Shoham soon thereafter proposed Manipulator [20], which is similar to MetaStrategy except that it answers question Q3 in a different way than MetaStrategy by initially selecting an outcome that gives it a higher payoff than does TFT. More sophisticated Builders, such as S++ [21], have mechanisms that allow them to continue to attempt a variety of desirable outcomes when the initially selected outcome fails, and hence possess greater potential to build consensus when associating with a broader set of associates.

By pursuing answers to different questions, Followers, Leaders, and Builders make different judgments about their associates, and then use these judgments differently. Followers attempt the extremely difficult task of deriving which strategies and algorithms their associates are currently using (Subtask 1), which might change over time in unexpected or difficult-to-observe ways. On the other hand, Leaders and Builders may not derive complex models of their associate's strategies or algorithms. They are largely ambivalent about which strategies or algorithms their associates use, but focus instead on finding a desirable end-goal their associate will likely accept. To find this end goal, they need a general understanding of what their associate might pursue (Subtask 1), and subsequently, what kinds of behaviors might generally influence them in that pursuit (Subtask 2). Answers to these questions do not typically change rapidly over time or scenarios, and there are successful rules-of-thumb readily available for finding them. Builders additionally model (verify) whether or not their associate is conforming with the intended outcome. Thus, these distinct judgments about their associates are often computationally less difficult and (I postulate) more easily generalize to a variety of scenarios than those formed by Followers.

So which genre of algorithm performs best? Prior work gives us more insights. In Table 1, I categorize a selected set of existing algorithms based on the questions they most prominently pursue. Simultaneously, I rank the algorithms based on their performance in a previously published study comparing those same 25 algorithms [17]. In that study, all 25 algorithms were paired with each other in many different games (with variations in payoff structure and game length) and were evaluated with respect to six different metrics. Both the overall rankings of the algorithms as well as the range of rankings (for specific selections of game categories and pairings with kinds of algorithms) are also given.

A variety of interesting observations can be made from Table 1. First, as indicated by the range of rankings of each algorithm, nearly all of the algorithms were a high performer in one or more scenarios. However, most of the algorithms also performed poorly in at least one scenario. Second, most algorithms selected in the study were Followers which focus primarily on answering questions Q1 and Q2 (either directly or indirectly). I state (without proper substantiation) my belief that this sampling of algorithms is reflective of the distribution of algorithms from the broader literature on repeated games – Followers appear to me to have been the most common form of algorithm designed by researchers to date. Third, despite the tendency to develop Followers, these algorithms were not the highest performers overall according to this evaluation of algorithms. The two highest performing algorithms, with respect to both overall ranking and best worst-case performance, were both Builders.

Although both of the highest-performing algorithms in this comparison of algorithms were Builders, Builders were not universally high performers either. MetaStrategy finished in the middle of the pack, substantially below its close cousin Manipulator. Given the large amount of Followers in this pool of algorithms, Bully was a more effective strategy than TFT, and hence Manipulator (which initially selects the same outcome as Bully) was a more effective Builder than MetaStrategy (which selects the same outcome as gTFT) overall. This was because it does not pay to be fair against Followers: one can

Table 1
The performance of and primary questions pursued by various algorithms in repeated games as implemented and evaluated by Crandall et al. [17]. *Overall Rank* is the average ranking of the algorithms (with 1 being the best) across all pairings, game categories, and considered game lengths, and measured with respect to six different performance metrics. *Rank Range* shows the best and worst ranking of each algorithm across all scenarios (algorithm categories, game categories, game lengths, and performance metrics). Q1 was checked only for algorithms that explicitly (and not implicitly) model their associate's strategy. A smaller checkmark indicates that the algorithm's mechanisms are perhaps primarily focused on the question, but do not fully carry it out. See Appendix A for further details on algorithm labels, implementations, and rankings.

Algorithm	Overall rank	Rank range	Q1	Q2	Q3	Q4	Label
S++	1	1 – 3			✓	✓	Builder
Manipulator	2	1 – 11			✓	✓	Builder
Bully	3	1 – 20			✓		Leader
S++/simple	4	1 – 23			✓	✓	Builder
S	5	1 – 18				✓	Builder?
Fict. Play	6	1 – 24	✓	✓			Follower
MBRL-1	7	1 – 14	✓	✓			Follower
EEE	8	1 – 16		✓			Follower
MBRL-2	9	1 – 19	✓	✓			Follower
Mem-1	10	2 – 21			✓		Leader
M-Qubed	11	1 – 22		✓			Follower
Mem-2	12	3 – 25			✓		Leader
MetaStrategy	13	1 – 24			✓	✓	Builder
WoLF-PHC	14	5 – 20		✓			Follower
QL	15	1 – 21		✓			Follower
gTFT	16	1 – 24			✓		Leader
EEE/simple	17	4 – 24		✓			Follower
Exp3	18	4 – 25		✓			Follower
CJAL	19	5 – 25		✓			Follower
WSLS	20	2 – 25			✓		Leader
GIGA-WoLF	21	9 – 24		✓			Follower
WMA	22	5 – 23		✓			Follower
Stoch. Fict. Play	23	3 – 25	✓	✓			Follower
Exp3/simple	24	7 – 24		✓			Follower
Random	25	20 – 25					None

just bully them. In effect, Manipulator initially answers question Q3 more effectively than MetaStrategy given this pool of associates, and both algorithms then retreat to a Follower strategy when the initially selected outcome is not followed by its associate (neither algorithm pursues multiple answers to question Q3).

On the other hand, S++ [21], a Builder algorithm which ranked highest in this study, iterates between questions Q3 and Q4 until it finds an outcome acceptable to both itself and its associate. This allows it to perform well in situations in which the outcomes selected by Bully, gTFT, and other strategies are eventually accepted by the associate.

2.3. Reflection

When an autonomous agent interacts with other agents, the judgments it makes (i.e., the models it creates) about its associates dictate its behavior. An autonomous agent can choose to make a variety of different kinds of judgments. One approach, which is perhaps predominant in the literature, is to try to estimate its associates' strategies either directly or indirectly, and then act to maximize utility with respect to these estimates. However, when associates are sophisticated, deriving or identifying their strategies is difficult, which often leads to incorrect or myopic models that produce inefficient results. On the other hand, evidence suggests that different, seemingly simpler, kinds of judgments about one's associates can be more effective. Rather than trying to model an associate's complete strategy, one can simply determine what desirable end outcome an associate is likely to accept (Subtask 1), play strategies to promote that outcome (Subtask 2), and continually model (verify) whether the associate accepts that outcome (Subtask 1). In this way, autonomous agents can, like Lincoln, simultaneously pursue Subtasks 1 and 2. At least some evidence presented in this section suggests that this form of agent modeling can be more effective.

In drawing this conclusion, it is appropriate to remark that these results should not be considered definitive. The results presented herein come from a single kind of scenario (repeated general-sum games), and only consider a small (yet non-trivial) sample set of algorithms. It may be that other algorithms that prioritize and address other forms of questions, or approach the same questions differently, will be more successful.

Despite these delimitations, I believe that these results are sufficient to urge caution against the prevalent algorithmic choice used by Followers, which is to maximize expected utility given current model estimates. In conformance with the sage advice (quoted at the beginning of this section) calling for people to first judge themselves before judging others, I argue that successful autonomous agents should first focus on ensuring that their own behavior will lead to desirable

Table 2
Properties of binding and non-binding communication signals.

Binding signals	Non-binding signals
· Tend to require commitment, as issuing these signals results in a change to world state, making them more difficult to undo	· Do not require commitment, as they do not change world state
· Tend to refer to current behavior	· Can refer to past, current, or future behavior
· Have lower bandwidth	· Have higher bandwidth: Complex and possibly unobservable strategies can be more easily conveyed

outcomes, and only then worry about whether or not their associates' behaviors also pursue those outcomes. When it comes to autonomous agents, judgments focused first on finding a mutually desirable end-goal can be both more simple, more successful, and more robust than those based on classifying the current strategies of other agents.

3. To signal or not to signal

> *We have two ears and one mouth so that we can listen twice as much as we speak.*
>
> [Epictetus, *The Golden Sayings of Epictetus*]

Broadly speaking, biological and artificial agents can derive models of each other using two forms of signals: signals that impact the world in some way (e.g., actions) and those that do not (e.g., words and gestures). I refer to these signals, respectively, as *binding* and *non-binding* signals. These signals have different properties (Table 2). Binding signals more fully signal commitment, as they require an agent to actually do something, though the amount of information that can be communicated with such signals is sometimes quite limited. On the other hand, non-binding signals may not necessarily denote commitment (since their only purpose is to change the "mental" states of others), but more information can often be communicated through them than through binding signals. Furthermore, whereas binding signals typically provide information about what an agent will do in the moment, non-binding signals can more easily explain both current and future behavior. These distinct properties make both forms of signals important in interactions among autonomous agents.

Despite the importance of both signal types in agent interactions, as well as the advanced work on agent communication (e.g., [22,23]) and automated negotiation (e.g., [24–26]), many sub-communities studying autonomous agents that interact with other agents largely consider only scenarios and solutions that support binding signals. For example, with just a few exceptions (e.g., [27,17,28]), the multi-agent learning community does not consider non-binding signals in their scenarios. Furthermore, research in *Ad hoc teams* [29,30] calls for autonomous agents to be able to learn to coordinate with each other without prior knowledge of each other or prior coordination. A popular interpretation of this agenda, including the interpretation given in the survey by Albrecht and Stone [2], is that this precludes the ability of the agents to communicate with each other via predefined signaling protocols. While vocabularies for communicating via non-binding signals can potentially be learned over time (e.g., [31]), the preclusion of predefined shared-communication protocols limits the use of non-binding signals in most Ad hoc teams. The majority of other works cited in Albrecht and Stone's survey likewise do not use non-binding signals in forming agent models.

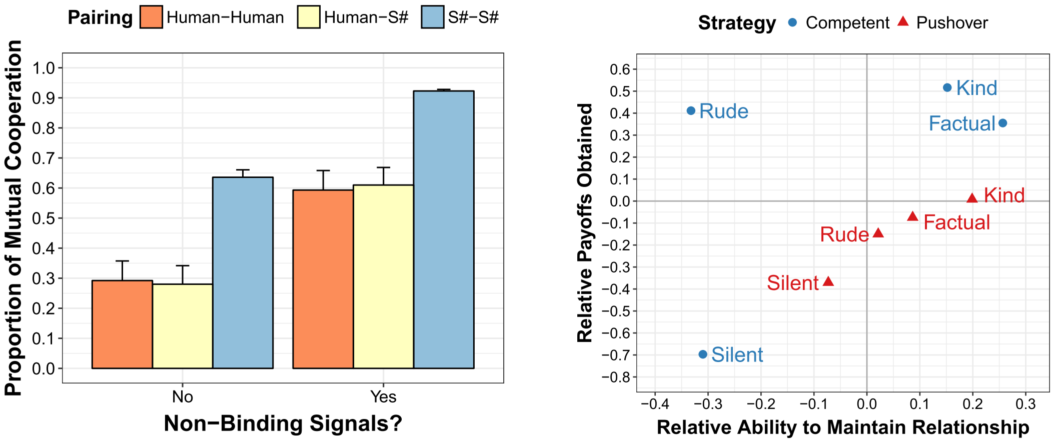
While having the capability to collaborate with other autonomous agents without shared-communication capabilities is desirable and challenging, it is a far more restrictive setting than is typically required of humans when they interact with each other. Even when people begin interacting with each other as complete strangers, they use non-binding signals to coordinate their behaviors even when they do not share the same spoken language. Indeed, such signaling is an innate human behavior [32], and has been shown to be important to coordination and compromise [33,34].

In this section, I argue that it is critical that autonomous agents consider and leverage predefined communication protocols when modeling other agents. As such, research into developing algorithms that better utilize non-binding communication signals should be more abundant. However, before developing these arguments, however, I believe it is appropriate to acknowledge why non-binding communication signals have not, to date, been given as broad of consideration in agent modeling as I believe they should.

3.1. Why not consider non-binding communication signals?

A hesitation to utilize non-binding communication signals in agent modeling and other forms of agent interactions has, I believe, been driven by a number of reasonable considerations:

- *Communication can be risky*. In domains in which the agents do not share all of the same preferences, there is reason to believe that one's signals will be used against them. This is certainly true in fully competitive domains (wherein non-binding communication should, in theory, convey no meaning at all), but may also be true (depending on the disposition of one's associates) when all agents' preferences are neither fully aligned nor fully in conflict. Additionally, even communicating with a teammate that shares one's preferences can be risky since the communication may be



(a) (b)

Fig. 3. The ability to send and receive non-binding signals is critical to establishing profitable cooperative relationships with other agents (including people) as demonstrated by the performance of the algorithm S# (which extends S++ with the ability to communicate via non-binding communication signals). (a) The proportion of mutual cooperation achieved by human-human, human-S#, and S#-S# pairings in a recently published study [17] (from which the figure is adapted) across several different games with and without the ability to send and receive signals. (b) In a separate study [28], different signaling personas (defined by the speech messages they sent to their partner) were paired with both a more and a less competent algorithm for selecting actions. The resulting eight agents were evaluated with respect to payoffs received and the ability to maintain a good relationship with people. Values are given as a function of the standard deviation from the mean. For details, see Oudah et al. [28] (from which the figure was adapted).

intercepted and used by other agents who are not friendly. For example, communicating to a teammate in robot soccer could result in the other team also observing the communication and thus thwarting plans.

- *Actions speak louder than words.* Since agents commit themselves to actions when using binding communication signals, these signals can be trusted more fully than non-binding signals.
- *Imbuing robots with advanced signaling capabilities is difficult.* General advanced signaling capabilities, such as gesture generation and recognition, facial signaling and recognition, and natural language generation and understanding are difficult and expensive to develop. Getting these signaling capabilities so that they work effectively in arbitrary scenarios (environments and tasks) is extremely difficult.
- *Expecting other autonomous agents to have shared communication capabilities is unlikely given current technologies.* When paired with arbitrary artificial agents, it is difficult to determine whether these other agents will have the appropriate hardware, algorithms, and assumptions that will allow communication to occur. For example, an autonomous agent may be uncertain whether its associate has a visioning system that can observe gestures, let alone that its associate will properly interpret them.
- *Adding communication may add complexity.* The need to keep track of the signals that have been exchanged among the agents may substantially (even exponentially) increase the state space over which an agent must reason.

While these reasons hold validity, the potential value of better using non-binding communication signals often outweighs these challenges. Additionally, recent developments in AI are mitigating some of these challenges. For example, progress in chat bots and natural language processing and generation make it increasingly likely that autonomous agents of the future will have predefined shared-communication protocols at their disposal. As such, I believe that now is the time to more fully consider non-binding communication in agent-modeling algorithms.

3.2. Illustrative example: non-binding signals can help an autonomous agent interact with others

While communication via non-binding signals has not been the norm in much prior work on multi-agent learning, it has been the focus of some prior work in this area (e.g., [27,17,28]). I personally learned the importance of such signals the hard way. Beginning as grad student, I sought to create algorithms that could effectively interact with people in repeated games. Though these algorithms had some success when paired with other algorithms, each time I tried to pair people with my algorithms, they failed to produce satisfactory results. After one particular failure (after more than a decade of trying, I am ashamed to say), the obvious answer finally hit me over the head like a ton bricks: *Why doesn't the dumb thing just talk*?

Once I came to this realization, my students and I were able to quickly extend S++ to form a new algorithm (S#) that quickly learns to cooperate with people in repeated games with cheap talk [17]. Rather than the typical setup used in most AI research on repeated games, repeated games with cheap talk allow players to each send a message to their associate prior to acting in each round. Fig. 3a, which shows sample results from one study we conducted, illustrates that Human-Human, Human-S#, and S#-S# pairings can and do substantially improve their ability to cooperate with each other (and in turn improve their individual payoffs) when they effectively use non-binding communication signals, even when the players do not share the same preferences.

Sending and receiving non-binding signals can have as great of an impact on an agent's performance as the behavioral algorithm it uses to select actions. For example, in a recent study conducted by Oudah et al. [28], people were paired with eight different artificial agents. These agents used two different algorithms for selecting actions, which I refer to as *Competent* (due to its ability to obtain higher payoffs in interactions with other artificial agents than the other algorithm) and *Pushover* (due to its tendency to get exploited). These two algorithms were combined with four different kinds of signaling personas: *Silent* (which did not send any messages to its associate) and three signaling personas that sent messages with different tones (which I refer to here as *Kind*, *Factual*, and *Rude*).

Fig. 3b summarizes the average performance of the eight resulting agents when they were paired with people in four repeated games. Performance is plotted with respect to two metrics: payoffs achieved and ability to engage people in the relationship. Agents that utilized the more competent algorithm to select actions achieved higher payoffs than agents that used the less competent algorithm except when *Competent* was paired with the *Silent* signaling persona. Interestingly, this competent but silent agent performed the worst overall with respect to both performance metrics, thus illustrating that, at least when paired with autonomous agents known as humans, the ability to utilize non-binding communication signals can be as important (or more so) than skillfully selecting actions.

These studies demonstrate just how important the ability to send and receive non-binding signals can be to the performance of autonomous agents that interact with other agents (including humans). Thus, rather than focusing solely on action histories to coordinate behavior, autonomous agents must effectively process and use non-binding signals within the decision-making algorithms they employ.

3.3. How should we use non-binding signals in agent modeling?

In the absence of a shared-communication protocol, learning to produce and receive meaningful non-binding communication signals between autonomous agents is challenging. The non-stationary environment caused by the agents adapting to each other makes it difficult to connect signals with behavior. As a result, non-binding signals do not easily make a positive contribution to the interaction. In fact, the presence of these signals can add complexity to the environment – complexity which can grow exponentially with the diversity of signals considered. Hence, though it remains an important and interesting research area, learning to interact with other agents while learning to send and interpret signals without a foundation of a predefined shared-communication protocol is unlikely to produce satisfactory results in general.

Human development informs, I believe, a simpler design of autonomous agents that use non-binding signals to better model each other. Humans develop language capabilities in their first years of life. Later in life, language and other forms of communication then form a foundation for entering into complex relationships and collaborations with other people. The design of autonomous agents can follow a similar course: first learning (or being given) a shared-communication protocol and then using this shared-communication protocol to more easily model and be modeled.

The way that shared-communication protocols can be used by autonomous agents to model others is an important open research question. By and large, the way that signals are used will reflect the kind of judgments pursued and utilized by the agents. Sen et al. [27] explored the effectiveness of a simple mechanism for use by Follower agents. In this work, an agent could simply choose whether or not to reveal which action it was going to take. For certain kinds of scenarios, this simple use of non-binding signals helped the agents form models of each other that led to convergence to more mutually beneficial equilibrium conditions.

Given a shared-communication protocol, non-binding signals can also easily be used by Builder algorithms via a process I refer to as LiPGuV[3] (*Li*sten, *P*ropose, *Gu*ard, and *V*erify). With LiPGuV, the Builder algorithm uses the shared-communication protocol to *listen* to and *propose* outcomes. The agent uses these non-binding signals to help determine an answer to question Q3 (which desirable outcome is likely to be accepted by my associate?). Given the possibility that its associates may not be honest or there may be misunderstanding about the shared-communication protocol, a Builder agent using LiPGuV then *verifies* that its associate's actions conform with its non-binding signals as part of answering question Q4. If this verification process reveals that the associate does not conform with the non-binding signals, it begins to *guard* itself from being misled by ceasing to account for these non-binding signals as it models and is modeled by its associate going forward.

In a sense, LiPGuV is so simple it appears trivial and not novel. Nor does it fully utilize various aspects of advanced research on agent communication (e.g., [22,23]) and automated negotiation (e.g., [24–26]), which could lead to further improvements. Yet, as demonstrated by how extending S++ with LiPGuV (to produce S#) improved this agent's relationships with people and other like-minded artificial agents (see Fig. 3a), its impact can be profound.

Intriguing future work is needed to effectively integrate LiPGuV and other mechanisms for using non-binding signals in agent modeling with advanced AI algorithms traditional designed for taking actions. Example important ongoing and future work includes:

[3] I made up the acronym LiPGuV during the writing of this paper. It is intended to refer to the fact that, if agents choose to be governed by the non-binding signals they send, the exchange of non-binding signals provides a simple means for agents to model and to be modeled.

- Dealing with imperfect shared-communication protocols. While an autonomous agent may develop communication protocols (such as language and gestures) that are likely to be shared with other agents, imperfections in these protocols are likely. As such, it is important for agents to understand when non-binding signals are not being properly understood and to update and extend these protocols as they build relationships (e.g., [31]).
- Connecting non-binding signals to internal states and actions. For autonomous agents to properly use non-binding signals in a way that has meaning, they must be able to connect these signals with their internal states and actions. In the case of S++, the connection between these signals and the algorithm's internal state was easily hand-coded since each expert strategy used by the algorithm embodies a general and simple high-level ideal. However, additional work is needed for us to learn how to create these mappings for other forms of AI algorithms (such as deep-learning models). Continued progress in *Explainable AI* (e.g., [35–37]) and other related areas is necessary. While initial work in fully cooperative domains shows promise (e.g., [38,39]), tying spoken language into machine-learning algorithms remains an open research area.
- Communicating strategy in more sophisticated scenarios. While sophisticated scenarios often reduce at some level to simpler scenarios (such as repeated normal-form games), determining how to effectively communicate strategies at the appropriate level in these more sophisticated scenarios remains an open question.

3.4. *Reflection*

The ancient Greek stoic philosopher Epictetus has been quoted as saying: "We have two ears and one mouth so that we can listen twice as much as we speak." This statement is commonly used to teach that one should be willing to listen to others more readily than to voice their own mind. But the statement also unintentionally speaks directly to the design of autonomous agents that interact with others. Autonomous agents that cannot talk to their associates cannot easily model or be modeled, and thus are not likely to establish and maintain successful relationships. Future work should better leverage existing work on agent communication (e.g., [22,23]) and automated negotiation (e.g., [24–26]) and identify how autonomous agents can use (potentially imperfect) shared-communication protocols to model and be model so as to enhance their ability to interact with others.

4. The tale of the tape

> *The measure of success is not whether you have a tough problem to deal with, but whether it is the same problem you had last year.*
> [John Foster Dulles, Former U.S. Secretary of State]

I fear that progress in understanding how to design autonomous agents that interact with other agents has been slowed by our mechanisms of evaluation. Scenarios in which autonomous agents interact with other agents vary so widely that it is challenging to fully articulate and remember which conditions each algorithm considers [40,2]. This, in turn, makes it easy for both AI researchers and the general public to over- or under-generalize results, and difficult to gain a solid understanding of the field. Furthermore, since academic research is typically disseminated via academic papers, evaluations of algorithms are often limited by both page-length constraints and reviewer tendencies. In some instances, this pushes us to conduct very detailed, but narrow analysis, such as proving that a particular mechanism is optimal in a particular scenario under particular assumptions about the other agents in the environment, rather than conducting more comprehensive evaluations that fully expose both the strengths and weaknesses of algorithmic processes.

The range of rankings of the algorithms displayed in Table 1 provides an illustration of this phenomena. The table shows that most of the 25 algorithms perform relatively well in at least one scenario covered in the cited study. But doing well in one scenario does not imply broad applicability or competence. As expected and has been shown in other studies (e.g., [41]), most of the algorithms also performed poorly relative to the other algorithms with respect to at least one condition. In fact, *no free lunch theorems* guarantee this result [42]. While much work over the last several decades has focused on developing complex Follower algorithms, in this study, the highest performing Follower algorithm on average was Fictitious Play [43], the oldest (and probably the simplest) of these Follower algorithms. This suggests that Fictitious Play's mechanisms for modeling the strategies of others may, with all of its flaws, be more general than many newer and more complex modeling mechanisms. Narrow evaluations focused on particular scenarios often do not reveal this and other phenomena.

In my opinion, we need more broad and comprehensive evaluations of agent-modeling algorithms to help us identify the mechanisms, theories, and design paradigms for agent modeling that perform well in many situations we might care about. Since the way we evaluate our algorithms largely dictates the kinds of algorithms that are created, a concerted effort is needed to develop a comprehensive and diverse set of metrics and test cases for autonomous agents that interact with other agents.

4.1. *Are the metrics we are using sufficient for evaluating our algorithms?*

A variety of different metrics have been used to evaluate the capabilities of autonomous agents that interact with other agents. Table 3 lists four broad categories for these metrics: payoff objective, model accuracy, generalizability, and relationship management. Of these categories, metrics that measure payoff objectives are the most commonly used. These metrics

Table 3
Metric categories and example metrics for measuring autonomous agents that interact with other agents.

Metric category	Description	Example metrics
Payoff objective	Measures the degree to which an autonomous agent obtains material payoffs that match some performance standard	Security, convergence to NE, regret minimization, social welfare, empirical performance
Model accuracy	Measures the quality of model predictions made by an autonomous agent	Change detection, best-response optimality
Generalizability	Measures the ability of an autonomous agent to perform effectively in a variety of tasks and environments with diverse associates	Task generality, partner flexibility, noise fragility
Relationship management	Measures impressions made by an autonomous agent on its associates	Attraction, trust, engagement

evaluate the quality of the solutions achieved by an autonomous agent in interactions with other agents. A variety of performance standards have been proposed and used in prior work, including being secure (e.g., [44,20]), minimizing regret (also known as universal consistency) (e.g., [45–47,44]), converging to Nash equilibria (e.g., [44,48–50]), maximizing social welfare, and other performance-related standards (e.g., [51,41,52,53]). Effective metrics of payoff objective should strongly correlate with actual empirical performance, such that if an algorithm meets the specified standard to a greater degree than another algorithm, then it should also achieve higher payoffs in interactions with other agents [21].

Metrics evaluating the quality of the material payoffs obtained by autonomous agents indirectly measure the quality of the agent's models of other agents. Alternatively, researchers have directly measured the quality of the models created by the autonomous agents. For example, it has been shown that, under certain circumstances, Fictitious Play's assessment of its associate's strategies will converge in the limit to the empirical distribution of its associate's actions, though it may not accurately model the actual strategy played by its associate at any give time [44]. Additionally, prior work has studied the ability of an autonomous agent to detect changes in their associates' strategies [54,55]. These and other direct measures of model accuracy helps us to better understand when autonomous agents are likely to be effective.

The third category of metrics listed in Table 3 are metrics identifying how well the algorithmic processes of the autonomous agent generalize to a variety of different conditions, including differences in the game environment (task generality), associates' behaviors (partner flexibility), and robustness to noise and uncertainty in the environment. For example, in repeated games, partner flexibility has been inferred using (a) the average payoff obtained in round-robin tournaments in which the agent is paired with a variety of different players and (b) population share in evolutionary simulations using replicator dynamics.

Metrics assessing how well algorithmic mechanisms generalize to a variety of conditions seem fundamental to the design of autonomous agents that interact with other agents. Perplexingly, however, in clear contrast to work in other fields of AI and machine learning which have been more careful about identifying overfitting, research in multi-agent learning and other scenarios in which autonomous agents interact often fails to evaluate how well mechanisms generalize. Evaluations most often focus on payoff objectives and model accuracy in specific scenarios. I believe this has limited the development of the field and should be corrected. To better understand which agent-modeling mechanisms best generalize and which do not, broader evaluations developing and considering explicit metrics of task generality, partner flexibility, etc. are needed.

Much of the analysis of autonomous agents that interact with other agents, assumes that the agents are forced to interact with each other, and vice versa. Albrecht and Stone [2] call these systems *closed* multi-agent systems. In reality, autonomous agents often operate in *open* multi-agent systems in which they can choose who they interact with. In such situations, autonomous agents must not only consider their payoffs, but they must also consider the impressions they make on other agent. Metrics in the category of *relationship management* measures these impressions. For example, the metric of *attraction* [28] measures the degree to which other agents (including people) would choose to interact with the autonomous agent given the choice. Metrics of relationship management have not been used widely in the broader AI community, though they are more often used in studies of human-robot interaction (e.g., [56]).

In short, while a variety of different metrics have been used extensively, several categories of metrics have been underutilized in the evaluation of autonomous agents that interact with other agents. In my opinion, to better understand which kinds of algorithmic mechanisms best allow autonomous agent to interact with other agents, we, as a community, need to expend more effort designing and utilizing metrics related to generality and relationship management.

4.2. Do we have sufficiently broad and valid test cases?

In addition to broadening the set of metrics we use to evaluate how well autonomous agents interact with other agents, I also believe that we need to do a better job applying a broader set of benchmarks and test cases to the algorithms that we develop. A typical paper published in our most competitive conference venues consists of developing an algorithmic mechanism, identifying a few theoretical properties of the algorithm, and then demonstrating superior empirical performance in one to three different games. Very often, these games are limited to a handful of scenarios, including prisoner's dilemmas, rock-paper-scissors (or matching pennies), simple grid world games such as predator-prey games, or an extensive-form game such as Poker.

While evaluations in a few domains is better than one domain, such evaluations are often insufficient to identify both the strengths and weaknesses of algorithmic mechanisms. Broader, more comprehensive, understanding of our algorithms is needed. For example, though prisoner's dilemmas are relevant and (for some reason) still extremely interesting, I argue that whole disciplines have been jaded by homogeneous evaluations of cooperation in that domain. Furthermore, world-class algorithms for playing Poker tend to minimize regret [57], but the rankings shown in Table 1 indicate that minimizing (particular definitions of) regret is often less successful in general-sum games. To more quickly identify such nuances, more comprehensive understanding of our algorithms is needed.[4]

Some comprehensive test cases have been developed already. For example, a sequence of work has resulted in a taxonomy of two-player normal-form games [59–62,18]. Several past works have used this taxonomy as a set of test cases that help us to identify the strengths and weaknesses of different algorithmic approaches (e.g., [41,63,17]). Furthermore, while typically not evaluated together, many different grid-world games have been established throughout the literature that potentially could be collected, classified, and used as a test-bed. In my opinion, evaluations in a broader set of scenarios would paint a much clearer picture of the strengths and weaknesses of algorithms than we get from evaluating algorithms in only a few games.

Competitions also provide useful test cases, as they allow us to better compare many algorithmic approaches in a handful of scenarios. These competitions have included a variety of zero-sum games such as Chess, Poker, Go, rock-paper-scissors, and general game-playing competitions. Other competitions have included various forms of robot soccer, prisoner's dilemma tournaments (patterned after Axelrod's initial tournaments), the trading agent competition, and the Lemonade-stand game [64]. Like the role of the international planning competition [65] in the development of planning algorithms, these competitions help facilitate agent modeling. However, care must be taken to not focus universally on any one competition in making general claims about how autonomous agents should be designed to interact with other agents.

4.3. Using benchmarks with caution

While I believe that we should use a broader set of metrics and test cases when evaluating how well autonomous agents model and interact with other agents, I also believe that we should use these tools with some caution. If over-utilized or not properly formed, benchmarks can cause us to over-focus on some problems more than others. Additionally, exorbitant expectations can arise causing us to reject modeling methods and algorithms with high potential, but that do not initially perform effectively on all benchmarks and metrics. Thus, care must be taken to ensure that benchmarks and metrics are properly utilized.

As an example, I use my own experience, as a graduate student, in developing and analyzing the algorithm M-Qubed [51]. This model-free reinforcement learning algorithm finished in the middle of the pack in the overall rankings shown in Table 1. M-Qubed has reasonably good asymptotic performance in repeated games, but it often takes tens of thousands of rounds before it converges. Thus, its performance is rather poor in shorter repeated games, but reasonably good in interactions of very long duration (particularly, as is customary of Follower algorithms, when paired with memory-bounded associates that do not adapt). Despite the spotty results, I learned much from creating and analyzing this algorithm, presenting it at a conference, and observing people's reactions and discussions about it. At least for me personally, these experiences had great qualitative value that I may not have had if the community had insisted that this algorithm performed well in all circumstances before it were published.

Thus, rather than serve as a litmus test for publication, my hope is that we will use a broader set of metrics and test cases as a means of more thoroughly understanding the strengths and weaknesses of the algorithmic mechanisms we study.

5. Conclusion

As indicated in Albrecht's and Stones' recent survey article on autonomous agents modeling other agents [2], research on agent modeling is now, in many senses, quite advanced. Nevertheless, repeated drawbacks to the various approaches of agent-modeling appear to prohibit the application of at least some of these methods broadly, particularly in scenarios in which autonomous agents interact with other adaptive agents. I believe that these challenges should cause us to reflect on both the kinds of judgments that autonomous agents should make about the agents they interact with, and the kinds of signals that are required to create these models in realistic interactions. In this reflection, I have argued that many scenarios call for autonomous agents to (a) model and clearly pursue desirable and acceptable outcomes rather than trying to predict the actual strategy being used by other agents and to (b) better utilizing non-binding signals (such as words and gestures) within that process. In so doing, I believe that we will be able to design autonomous agents that better interact with other agents (including people) in arbitrary environments.

[4] A call for a broader set of test cases for evaluating autonomous agents as they interact with other agents is not new. Shoham and associates established GAMUT [58], and similarly called for more systematic evaluations more than a decade ago [40]. However, our response has been sluggish. Editorial note: Perhaps our publishing culture, which promotes rapid dissemination of results through smaller conference papers rather than more refined journal articles, and our paper-reviewing tendencies are partially to blame for this.

Furthermore, to more fully develop and analyze agent-modeling methods, I believe that the community as a whole should make a concerted effort to develop a broader set of metrics and to then more comprehensively evaluate methods using a broader set of test cases. Given the broad nature of autonomous agents that model other agents, such metrics and test cases are necessary to understand when each paradigm is likely to be successful. In other words, perhaps all we lack is slightly different judgments than we are prone to make, combined with mouths, ears, and a little more tape.

Declaration of competing interest

The author declares no conflict of interested associated with this publication, including no significant financial support that could have influenced its outcome.

Acknowledgements

This work was supported in part by the U.S. Office of Naval Research under Grant #N00014-18-1-2503. All opinions, findings, conclusions, and recommendations expressed in this paper are those of the author and do not necessarily reflect the views of the Office of Naval Research.

Appendix A. Categorizing and ranking algorithms

Table 1 presents both a categorization of selected algorithms with respect to the questions they primarily pursue as well as a ranking of these algorithms. This categorization and rankings were based on implementations of the algorithms as described by Crandall et al. [17]. I refer to the reader to Supplementary Note 3 of that work for details. In this appendix, I describe how I categorized these algorithms with respect to questions Q1–Q4 (see Section 2). I also overview how the algorithms were ranked, though details of these ranking are given by Crandall et al. [17].

A.1. Categorizing algorithms

Each algorithm in Table 1 is categorized with respect to what I determine to be its *primary* pursuits. As is the case with many classifications, categorizing algorithms with respect to these questions can be somewhat messy and subjective. This description is intended to make these classifications transparent.

To categorize the algorithms, I asked the following four questions (each corresponding to one of the four questions discussed in Section 2.2) about each algorithm:

- *To determine a match to question Q1.* Does the algorithm explicitly try to determine its associate's strategy or algorithmic mechanisms, and does it use this determination as a primary means for selecting its actions and strategies?
- *To determine a match to question Q2.* Does the algorithm typically select its own actions and strategy as a response to either an explicit or implicit model estimate of its associate's strategy or algorithmic mechanisms?
- *To determine a match to question Q3.* Does the algorithm commit (for at least some period of time) to an outcome, or to playing a strategy focused on a particular outcome, that there is justified reason for believing (a priori) that it will be desirable to both itself and its associate?
- *To determine a match to question Q4.* Does the algorithm verify whether the outcome (or an outcome that yields equivalent or sufficient quality) it committed to is being played by its associate?

For each question, if the answer was *yes*, then I generated a checkmark in Table 1 with respect to the corresponding question for that algorithm. For the few cases in which I felt the answer was *somewhat* but not a sufficiently resounding *yes*, I produced a smaller checkmark in the table.

Some algorithms pursue all of the four questions (Q1-Q4) to some degree. For example, some Builders (e.g., MetaStrategy, Manipulator, and S++) do model their associate's strategies, but they typically only respond to these models after attempts at reaching desirable outcomes (answers to question Q3) have failed. Thus, I did not consider that these are primary questions pursued by these algorithms.

A.2. Ranking algorithms

Crandall et al. [17] compared the 25 algorithms listed in Table 1 by pairing the algorithms together in a comprehensive set of different repeated games using the taxonomy developed by Bruns [18], and evaluating their performance with respect to six different metrics in these repeated games. Given the lack of a single metric by which to evaluating algorithms in repeated games, Crandall et al. used a suite of metrics to compare these algorithms. This suite of algorithms included:

- The average payoff obtained by the algorithms across all pairings, games, and rounds.
- The proportion of associates against which the algorithm performed best (compared to all of the other 25 algorithms).
- The average payoff obtained when paired with its worst-case associate.

- The usage rate over 10,000 generations of application of the replicator dynamic.
- Two separate forms of *elimination tournaments*.

For each metric, the algorithms were compared in different scenarios. These scenarios were categorized based on the payoff structure of the games played (a total of 720 2 × 2 matrix games were considered), the length of the repeated game, and the type of associate. The *Overall Ranking* in Table 1 was based on rankings averaged over all metrics, payoff structures, game lengths, and associates. The *Rank Range* was determined as the best and worse rank obtained for all categories of games, game lengths, and associates.

Further details about the metrics, game groupings, and algorithm groupings are given by Crandall et al. [17] (see primarily Supplementary Note 3).

References

[1] D.K. Goodwin, Team of Rivals: The Political Genius of Abraham Lincoln, Simon & Schuster, New York, 2012.

[2] S.V. Albrecht, P. Stone, Autonomous agents modelling other agents: a comprehensive survey and open problems, Artif. Intell. 258 (2018) 66–95.

[3] A. Grant, Give and Take: A Revolutionary Approach to Success, Penguin Books, 2013.

[4] J. Nachbar, Prediction, optimization, and learning in repeated games, Econometrica 65 (2) (1997) 275–309.

[5] D. Foster, H. Young, On the impossibility of predicting the behavior of rational agents, Proc. Natl. Acad. Sci. 98 (22) (2001) 12848–12853.

[6] J. Nachbar, Beliefs in repeated games, Econometrica 73 (2) (2005) 459–480.

[7] R. Axelrod, The Evolution of Cooperation, Basic Books, New York, 1984.

[8] M.L. Littman, P. Stone, A polynomial-time Nash equilibrium algorithm for repeated games, Decis. Support Syst. 39 (2005) 55–66.

[9] R. Powers, Y. Shoham, New criteria and a new algorithm for learning in multi-agent systems, in: Neural Information Processing Systems, 2004, pp. 1089–1096.

[10] E.M.D. Cote, M.L. Littman, A polynomial-time Nash equilibrium algorithm for repeated stochastic games, in: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, 2008, pp. 419–426.

[11] M. Elidrisi, N. Johnson, M. Gini, J.W. Crandall, Fast adaptive learning in repeated stochastic games by game abstraction, in: Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems, 2014.

[12] J. Foerster, R.Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, I. Mordatch, Learning with opponent-learning awareness, in: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, 2018, pp. 122–130.

[13] W. Shen, A.A. Khemeiri, A. Almehrzi, W.A. Enezi, I. Rahwan, J.W. Crandall, Regulating highly automated robot ecologies: insights from three user studies, in: Proceedings of the 5th International Conference on Human-Agent Interaction, 2017.

[14] M.L. Littman, P. Stone, Leading best-response strategies in repeated games, in: IJCAI Workshop on Economic Agents, Models, and Mechanisms, Seattle, WA, 2001.

[15] M. Nowak, K. Sigmund, A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game, Nature 364 (1993) 56–58.

[16] W.H. Press, F.J. Dyson, Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent, Proc. Natl. Acad. Sci. USA 109 (26) (2012) 10409–10413.

[17] J.W. Crandall, M. Oudah Tennom, F. Ishowo-Oloko, S. Abdallah, J.F. Bonnefon, M. Cebrian, A. Shariff, M.A. Goodrich, I. Rahwan, Cooperating with machines, Nat. Commun. 9 (233) (2018).

[18] B.R. Bruns, Names for games: locating 2 × 2 games, Games 6 (4) (2015) 495–520.

[19] F.R. Covey, The 7 Habits of Highly Successful People: Restoring the Character Ethic, Simon & Schuster, New York, 1989.

[20] R. Powers, Y. Shoham, Learning against opponents with bounded memory, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, 2005, pp. 817–822.

[21] J.W. Crandall, Towards minimizing disappointment in repeated games, J. Artif. Intell. Res. 49 (2014) 111–142.

[22] Y. Shoham, K. Layton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Cambridge University Press, New York, NY, 2008.

[23] A.K. Chopra, M.P. Singh, Agent communication, in: G. Weiss (Ed.), Multiagent Systems, 2nd edition, MIT Press, Cambridge, MA, 2013, pp. 101–142, Ch. 3.

[24] F. Lopes, H. Coelho (Eds.), Negotiation and Argumentation in Multi-Agent Systems: Fundamentals, Theories, Systems and Applications, Bentham Books, 2014.

[25] S. Fatima, S. Kraus, M. Wooldridge, Principles of Automated Negotiation, Cambridge University Press, 2014.

[26] Automated negotiating agent competition 2018, https://www.ijcai-18.org/anac/index.html. (Accessed 6 December 2019).

[27] S. Sen, S. Airiau, R. Mukherjee, Towards a Pareto-optimal solution in general-sum games, in: Proceedings of the 2nd International Conference on Autonomous Agents and Multi-Agent Systems, 2003, pp. 153–160.

[28] M. Oudah, T. Rahwan, T. Crandall, J.W. Crandall, How AI wins friends and influences people in repeated games with cheap talk, in: Proceedings of the 32nd National Conference on Artificial Intelligence, 2018.

[29] M. Rovatsos, G. Weiß, M. Wolf, Multiagent learning for open systems: a study in opponent classification, in: Adaptive Agents and Multi-Agent Systems, in: LNAI, vol. 2636, Springer, 2003, pp. 66–87.

[30] P. Stone, G.A. Kaminka, S. Kraus, J.S. Rosenschein, Ad hoc autonomous agent teams: collaboration without pre-coordination, in: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010, pp. 1504–1509.

[31] P. Chocron, M. Schorlemmer, Vocabulary alignment in openly specified interactions, in: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems, 2017.

[32] A. Pentland, To signal is human, Am. Sci. 98 (2010) 204–211.

[33] D. Sally, Conversation and cooperation in social dilemmas a meta-analysis of experiments from 1958 to 1992, Ration. Soc. 7 (1) (1995) 58–92.

[34] D. Balliet, Communication and cooperation in social dilemmas: a meta-analytic review, Ration. Soc. 54 (1) (2009) 39–57.

[35] M. van Lent, W. Fisher, M. Mancuso, An explainable artificial intelligence system for small-unit tactical behavior, in: Proc. of the 16th Conference on Innovative Applications of Artificial Intelligence, 2004.

[36] M.G. Core, H.C. Lane, M. van Lent, D. Gomboc, S. Solomon, M. Rosenberg, Building explainable artificial intelligence systems, in: Proc. of the 18th Conference on Innovative Applications of Artificial Intelligence, 2006.

[37] D. Gunning, Explainable artificial intelligence, Tech. Rep. DARPA-BAA-16-53, DARPA Broad Agency Announcement, August 2016, http://www.darpa.mil/program/explainable-artificial-intelligence.

[38] J.N. Foerster, Y.M. Assael, N. de Freitas, S. Whiteson, Learning to communicate with deep multi-agent reinforcement learning, in: Advances in Neural Information Processing Systems, 2016, pp. 2137–2145.

[39] H.J. Yoon, H. Chen, K. Long, H. Zhang, A. Gahlawat, D. Lee, N. Hovakimyan, Learning to communicate: a machine learning framework for heterogeneous multi-agent robotic systems, in: Proceedings of AIAA SciTech, 2019.

[40] Y. Shoham, R. Powers, T. Grenager, If multi-agent learning is the answer, what is the question?, Artif. Intell. 171 (7) (2007) 365–377.

[41] S.V. Albrecht, S. Ramamoorthy, Comparative evaluation of MAL algorithms in a diverse set of ad hoc team problems, in: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, 2012, pp. 349–356.

[42] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.

[43] G.W. Brown, Iterative solutions of games by fictitious play, in: T.C. Koopmans (Ed.), Activity Analysis of Production and Allocation, John Wiley & Sons, New York, 1951.

[44] D. Fudenberg, D.K. Levine, The Theory of Learning in Games, The MIT Press, 1998.

[45] D.P. Foster, R. Vohra, Regret in the on-line decision problem, Games Econ. Behav. 29 (1999) 7–35.

[46] M. Bowling, Convergence and no-regret in multiagent learning, in: Adv. Neur. In., 2004, pp. 209–216.

[47] A. Greenwald, A. Jafari, A general class of no-regret learning algorithms and game-theoretic equilibria, in: Proceedings of the 16th Annual Conference on Computational Learning Theory, 2003, pp. 2–12.

[48] J. Hu, M.P. Wellman, Multiagent reinforcement learning: theoretical framework and an algorithm, in: Proceedings of the 15th International Conference on Machine Learning, 1998, pp. 242–250.

[49] M.L. Littman, Friend-or-foe: Q-learning in general-sum games, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 322–328.

[50] M. Bowling, M. Veloso, Multiagent learning using a variable learning rate, Artif. Intell. 136 (2) (2002) 215–250.

[51] J.W. Crandall, M.A. Goodrich, Learning to compete, compromise, and cooperate in repeated general-sum games, in: Proceedings of the 22nd International Conference on Machine Learning, 2005.

[52] D. de Farias, N. Megiddo, Exploration–exploitation tradeoffs for expert algorithms in reactive environments, in: Adv. Neur. In., 2004, pp. 409–416.

[53] R. Arora, O. Dekel, A. Tewari, Online bandit learning against an adaptive adversary: from regret to policy regret, in: Proceedings of the 29th International Conference on Machine Learning, 2012, pp. 1503–1510.

[54] P. Hernandez-Leal, Y. Zhan, M.E. Taylor, L.E. Sucar, E. Munoz de Cote, Efficiently detecting switches against non-stationary opponents, Auton. Agents Multi-Agent Syst. 31 (4) (2017) 767–789.

[55] M. Ravula, S. Alkobi, P. Stone, Ad hoc teamwork with behavior switching agents, in: International Joint Conference on Artificial Intelligence, 2019.

[56] E. Short, J. Hart, M. Vu, B. Scassellati, No fair!! An interaction with a cheating robot, in: Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction, HRI, 2010, pp. 219–226.

[57] M. Bowling, N. Burch, M. Johanson, O. Tammelin, Heads-up limit hold'em poker is solved, Science 347 (6218) (2015) 145–149.

[58] E. Nudelman, J. Wortman, K. Leyton-Brown, Y. Shoham, Run the GAMUT: a comprehensive approach to evaluating game-theoretic algorithms, in: Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems, NYC, NY, 2004.

[59] A. Rapoport, M.J. Guyer, A Taxonomy of 2 × 2 Games, Bobbs-Merrill, 1967.

[60] A. Rapoport, M.J. Guyer, D.G. Gordon, The 2 × 2 Game, The Univ. of Michigan Press, 1976.

[61] S.J. Brams, A Theory of Moves, Cambridge University Press, 1994.

[62] D. Robinson, D. Goforth, The Topology of the 2 × 2 Games: A New Period Table, Routledge, 2005.

[63] S.V. Albrecht, J.W. Crandall, S. Ramamoorthy, An empirical study on the practical impact of prior beliefs over policy types, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 1988–1994.

[64] The lemonade stand game, http://martin.zinkevich.org/lemonade/. (Accessed 29 August 2019).

[65] International planning competition 2018, https://ipc2018.bitbucket.io/. (Accessed 29 August 2019).