September 2018

Markov Games for Persuasive Dialogue

Niklas RACH [a,1], Wolfgang MINKER [a] and Stefan ULTES [b]
[a] *Institute of Communication Engineering, Ulm University, Germany*
[b] *Department of Engineering, University of Cambridge, UK*

Abstract. This work discusses the formulation of argumentative dialogue as Markov game. We show how formal systems for persuasive dialogues that adhere to a certain structure can be reformulated as Markov games and thus be addressed as Reinforcement Learning task in a multi-agent setting. We validate our approach on an implementation of a proof of principle scenario where we show that the optimal policy can be learned.

Keywords. Persuasive Dialogue, Markov Games, Reinforcement Learning

1. Introduction

The ability of exchanging arguments about a certain topic or issue is essential in human conversation in order to resolve conflicts, exchange knowledge or persuade an opponent of ones point of view. Consequently, systems that are capable of interacting with humans and each others in the same way are of particular interest in view of tasks as tutoring, reasoning or deducing new knowledge. Despite the large amount of arguments present in various forms on the Internet, systems that are capable of exchanging arguments with human users (and each other) are scarce as they have to overcome different obstacles [1]. One of them is the development of a flexible system strategy that is competitive for humans. Within this work, we address this issue by formalizing the argumentation as Markov game [2] which allows to learn the agent strategy with Reinforcement Learning (RL) in a multi-agent setup. In contrast to existing approaches like the ones presented in [3, 4], no training corpus or pre-defined strategy is required as the optimal policy is learned solely from the structure of the problem and the available arguments. Moreover, RL allows to address large and complex scenarios in which the dimensionality makes analytical approaches impractical.

Following the classification of argumentative dialogue in [5], our focus lies on persuasive dialogue and a formalization of the same as dialogue game [6]. The latter ones model the interaction between agents as game in which each side is allowed to play different moves to manipulate and extend the present lines of argumentation. In addition to the straight forward exchange of arguments, these games generally also allow for strategical moves as for example questioning the validity of a move or retracting a previous statement.

We show how dialogue games for argumentation adhering to a general structure [6] can be formulated as Markov games. The approach is tested on a specific instantiation

[1] Corresponding Author: Niklas Rach; E-mail: niklas.rach@uni-ulm.de.

of such a game and a proof of principle scenario where we show that the optimal policy can be found by means of RL.

The remainder of this paper is as follows: Section 2 recalls the Markov game formalism and RL whereas the formal description of argumentation as dialogue game is covered in Section 3. In Section 4, we address the formulation of dialogue games for argumentation as Markov game. Our implementation and the respective results are presented in Section 5. Finally, Section 6 includes a conclusion and perspectives for future work.

2. Markov Games and Reinforcement Learning

In this section, we first recall the definition and important properties of the Markov game formalism. Afterwards we discuss basics of RL in this context including the applied algorithms.

2.1. Markov Games

Following the notation of [7], a discounted Markov game with I players can be described as tuple $(S, \mathbf{A}, \mathbf{r}, T, \gamma)$ with S the set of states or state space, $\mathbf{A} = A_1 \times ... \times A_I$ the joint action space of all agents that includes the (sub)sets of actions available to player i in the respective state $A_i(s^i)$ and γ a discount factor for future rewards. The reward function $r_i : S \times \mathbf{A} \to \mathbb{R}$ determines the real valued reward, given the current state s of the respective agent and the current actions of all players and is included in the joint reward function $\mathbf{r} = r_1 \times ... \times r_I$ of all players. The transition probability function $T : S \times \mathbf{A} \times S \to [0,1]$ determines the probability for reaching a state s', given the current state s and the actions of all players. In the special case of $I = 1$, this formalism corresponds to the formal description of a Markov decision process (MDP).

In a Markov game, each agent selects actions according to a policy function $\pi_i : S \to \mathscr{D}(A_i)$ that is a probability distribution over possible actions given some state s. As the optimal policy can be stochastic, it is formally a mapping from the set of states to the set of distributions over actions $\mathscr{D}(A_i)$. The goal of each agent is to find the policy that maximizes his expected discounted sum of future rewards

$$V_i(s, \pi) = \mathbb{E}(\sum_{t=k}^{\infty} \gamma^t r_i(s_t, \pi(s_t)) | s_k = s) \tag{1}$$

with $\pi : S \to \mathscr{D}(\mathbf{A})$ the joint strategy of all agent. V_i is called the value function of agent i. A joint policy π is defined to be optimal if each agents policy π_i is optimal with respect to the policies of all other agents. This is called a Nash equilibrium and discounted stochastic games were shown to posses at least one in stationary policies [8]. Throughout this work, we consider two-player games with altering turns, which ensures that an optimal deterministic policy exists [9].

2.2. Reinforcement Learning

RL provides a way of solving MDPs (and Markov games) in scenarios where the solution cannot be computed analytically, for example in the case of incomplete information

(transition and/or reward function unknown) or increasing dimensionality. Most work on RL is focused on the single agent case and single agent learners are thus well studied and can also be applied to Markov games [10]. Thus, we consider methods of this kind in the herein discussed scenario. Throughout this work, we focus on (model-free) value based methods that aim at approximating the optimal state-action (or Q) function. The latter one encodes the expected future reward given action a in state s and following policy π afterwards. For the single agent case it can be described in terms of the state value function in Eq. 1 as

$$Q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} T(s,a,s')V(s',\pi) \tag{2}$$

If the optimal Q-function is known, the optimal policy can be derived from this function as $\pi(s)^* = \text{argmax}_a Q(s,a)$. The Q-learning algorithm [11] approximates the Q-function during training by updating it according to

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(r(s,a) + \gamma \max_a Q(s',a)) \tag{3}$$

where s' denotes the state the agent is in after executing action a and α is called the learning rate. It should be noted that due to the *max* operator over actions this is an *off-policy* algorithm. The respective *on-policy* variation is usually referred to as SARSA - state action reward state action - algorithm [12] and replaces the action a identified by the *max* operator with the action a' chosen according to the current policy in state s'. It should be noted that one main issue of the Q-learning algorithm in the context of Markov games lies in the inability to learn stochastic policies [10]. However, as discussed earlier, the herein considered case of altering turns has an optimal deterministic policy and can thus be addressed by the algorithm. In order to efficiently deal with large state spaces, we employ linear function approximation [12] of the form $Q(s,a) = \sum_i w_i \phi_i(s,a)$ where w_i denote weights that are optimized during learning and ϕ_i features characterizing the present state-action pairs. As similar state-action pairs have similar features, this approximation offers an educated guess for the Q-function value of states that were not visited during training based on the values for features this state shares with others.

3. Formal System for Persuasive Dialogue

In this Section we recall the general definition of a formal system for persuasive dialogue (in short Argument Games) introduced by [6]. An Argument Game is described by the pair $(\mathscr{L}, D)$ with $\mathscr{L}$ a logic for defeasible argumentation [13] and D a so called *dialogue system proper*.
$\mathscr{L}$ encodes the arguments and their formal relations to each other that are available in the game. It consists of a formal language L_t, a set of inference rules R, arguments *args* that are elements of L_t connected by instantiations of rules in R and a binary relation between the elements of *args* that defines which arguments defeat each other. Thus it can be formally described as tuple $(L_t, R, args, \rightarrow)$. Throughout this work we say that $\mathscr{L}$ encodes the *argument structure* of the game.

The *dialogue system proper D* on the other hand encodes the rules of the game. It is a tuple (L_c, P, C), consisting of a communication language L_c, a protocol P structuring the

interaction between the participants (including to determine legal moves in each state) and a set of commitment rules C that regulate implications and restrictions arising from playing certain moves.

Elements of L_c have the form $\beta = p(c)$ where p is a dialogue act or performative, respectively out of a set $\mathscr{P}$ and $c \in L_t$ or $c \in args$ an element of the defeasible logic. Over L_c, two binary relations R_a and R_s are defined that encode *attacking* and *surrendering* replies.

The protocol P relies on moves m_k and dialogues $d = m_1, ..., m_k$ i.e. sequences of moves and determines the player(s) to move and a set of legal moves for each dialogue. In addition, it regulates termination, i.e. when the game is finished. Each move has the form $m_k = (k, \beta_k, pl(m_k), \tau(m_k))$ where k is the identifier of the move, β_k its speech act, $pl(m_k)$ the respective player and $\tau(m_k)$ the identifier of the target move, i.e. the previous move m_k responses to. In the scope of this work, let M denote the set of all moves and $M^{\leq\infty}$ the set of all possible dialogues. [14] distinguishes between single and multi-move protocols (whether the player to move switches after each move), single and multi-reply protocols (whether multiple responses to a move are allowed) and immediate- and non-immediate-response protocols.

The outcome of an Argument Game is defined by means of an outcome function or rule determining the winner of the game. An optimal strategy thus generally depends on the specific protocol, the available arguments and the winning criterion. As a consequence, the complexity of finding this optimal strategy generally depends on the size of the argument structure and structures over a certain dimensionality may thus require approximative methods like RL (depending on the specific protocol and winning criterion).

3.1. Prakken's Argument Game

As an example for the above introduced formalism we briefly discuss one particular game introduced in [15]. The communication language L_c of this game includes five types of moves which are *claim*, *argue*, *why*, *concede* and *retract*. The game is played by two players (A and B), where the stance of player A (proponent) is to defend the *claim* and the stance of player B (opponent) is to attack the same. Thus, player A always starts the game by introducing a *claim*.

The protocol P relies on a relevance criterion to determine whether or not a move can be addressed in the actual turn. To this end, a binary status is assigned to each played move at each state of the game, defining it as either *in* or *out*. If an attack on a certain move leads to a change of the status of the *claim*, it is a relevant target. Only relevant targets can be addressed in the actual turn. The player to move is determined by the status of the *claim* and switches once the status of the *claim* is switched. Thus, the protocol is multi-move, as each player has to play moves until the status of the initial claim is switched. It is also a multi-reply, as it is possible to respond to a move more than once. The game ends if the player to move has no legal move left and therefore is not able to change the status of the *claim* any more. This player loses the game.

This definition of winning leads to an intuitive optimal strategy for player B that is independent of the argument structure (which is generally not the case): If player B plays an attack move in each turn, player A will eventually run out of arguments and thus always lose the game. Despite the fact that this is a strategy that is not considered optimal amongst humans, it offers a baseline for the herein introduced formalism, which is why we choose this particular game for the numerical validation of our approach.

4. Persuasive Dialogue as Markov Game

In this section we present the main contribution of the paper of formulating the Argument Game (Sec. 3) as a Markov game. We stress that this formulation is possible for any game that has the above discussed elements and is not limited to a specific instantiation. The state at time t of agent i can generally be expressed in terms of the dialogue $d_t = m_1, ..., m_t$ and the current commitments of the respective agent C_t^i as

$$s_t^i = (d_t, C_t^i) \tag{4}$$

The set of possible actions for each agent in a certain state $A_i(s^i)$ can be derived from the protocol function $Pr : M^{\leq\infty} \rightarrow 2^M$ defined in the protocol of the Argument Game as $A_i(s_t^i) = Pr(d(s_t^i)))$ with $d(s_t^i)$ the dialogue encoded in state s_t^i. Here, 2^M denotes the power set of M. The available actions are thus equivalent to the legal moves in the Argument Game (see Eq. 3). In order to enable multiple-turn games, we assign the set of possible actions $A_i(s^i)$ to the agent if he is the player to move and a *wait* action a_w that does nothing otherwise.

The transition function T determines the state transitions given a state of an agent, his action and the action of the opponent. In the herein discussed case, it extends the dialogue in s by the respective moves $m_t = a_t$ associated with each agent's action and updates the commitment of agent i according to the commitment rules provided by the Argument Game. Throughout this work we consider a deterministic environment.

The reward of each agent can be derived from the outcome rule of the Argument Game. According to [14], an Argument Game of the kind discussed here is a zero-sum game meaning that whenever one participant wins, his opponent loses the game. A straight forward approach is thus to assign a reward of $+20$ to the winning agent and a negative reward of -20 to the losing one at the end of the game. However, the herein introduced formalization does not rely on the zero-sum assumption and can be applied to general-sum games as well by modifying the reward function appropriately. Finally, the discount factor can be chosen appropriately and was set to $\gamma = 0.9$ in the herein considered case.

It should be noted that this general formulation comes with a large state space that is impractical in view of implementations. We thus discuss a modification of the above introduced formalization that allows to decrease the dimensionality of the state space under the following condition:

If the legality of a move depends on the temporal order of previous moves, this dependency is restricted to the latest move.

Although it is not fulfilled by definition, we are not aware of a formal system violating this assumption. The most common systems in which the temporal order of arguments is relevant are the ones with immediate-response protocols and systems of this kind do not violate the above posed condition. Thus it is a rather weak restriction in view of implementations. If the discussed condition is fulfilled, the dialogue d_t in the state representation of equation 4 can be replaced by a graph $G = (g, e)$ with $g \subseteq L_c$ and the edges e given by the relations R_a and R_s defined between elements of L_c in the Argument Game. In order to enable immediate response-protocols, the latest speech act β_t is also included into the state, yielding the representation $s_t^i = (G_t, C_i^t, \beta_t)$ that summarizes similar states in the original formalization in one new state. In practice, the graph can be encoded in the respective adjacency matrix to be computationally efficient.

As a consequence, the set of possible action for each agent $A_i(s^i)$ needs to be modified as well. Given the new state space S, the set of all dialogues $M^{\leq\infty}$ and the set of all moves M it can be defined as

$$A_i(s_t^i) = \chi(Pr(\Delta(s_t^i))) \tag{5}$$

with $\Delta : S \rightarrow M^{\leq\infty}$ a function that maps the current state to a hypothetical dialogue, Pr the protocol function of the Argument Game discussed above and $\chi : 2^M \rightarrow 2^A$ a function that maps a set of moves to a set of actions. The actions now take the form $a_t = (pl(m_t), \beta_t, g_t)$ with g_t the node (i.e. speech act) in the graph G_t that a_t responds to.

The last required adjustment is in the transition function, that has to update the graph G_t instead of the dialogue by including the edges associated with the current move of each player.

5. Experiment and Results

In the following, we evaluate the above introduced approach in an experiment based on the specific argument game introduced in Section 3.1. This choice is due to the fact that the optimal policy is known for this particular instantiation and thus allows to validate the policies achieved by use of the Markov Game approach.

Learning was done via $Q(\lambda)$ and $SARSA(\lambda)$ algorithm with linear function approximation and an ε-greedy strategy (see [12] for a detailed discussion). It is important to note that the linear approximation generally allows to encode knowledge about the optimal strategy in the features. As our point is to show that this knowledge is not needed in order to optimize the strategy, we consequently do not explore this advantage. Similar to [16], one agent was assigned a fixed policy (*reference policy*) against which the other agent was trained (*training policy*). Both agents start with a random policy and after each n episodes, the *reference policy* is replaced by the actual *learning policy*. An episode consists of one closed dialogue or epoch of the Argument Game, respectively. In the following discussion, we denote a set of n training episodes as a super-iteration. The stance of each agent is determined randomly at the beginning of each episode in order to train a policy that is applicable to both sides. Both algorithms were trained on an overall of 40000 episodes with the reference policy updated after every $n = 4000$ episodes for 10 randomly generated argument structures with 10 argument components each. The latter ones were employed in order to ensure that the outcomes do not depend on one specific structure. The learning rate was set to $\alpha = 0.02$ and the exploration rate was 10%. After each episode, the current policy was evaluated with 0% exploration.

Figure 1 shows the thereby achieved average reward of the training agent when assigned to stance B (grey) and overall (blue) as a function of the number of super-iterations. As discussed earlier, a Nash equilibrium means that both strategies are optimal against each other, i.e. changing the policy does not yield an advantage for any side. As both agents start with the same policy every n episodes, an average reward around zero means that no improvement could be achieved by changing this policy further and both sides perform equally well. It should be noted that this is not a convergence guarantee since the training episodes or the considered amount of super-iterations might not be enough to achieve an improvement (as can be seen in Figure 1 between super-iteration

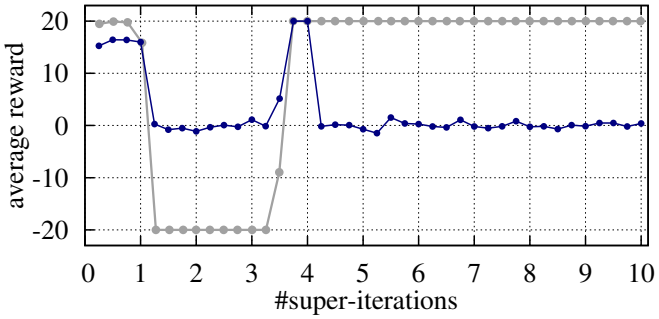

Figure 1. Average reward for the training agent as a function of super-iterations when assigned to stance B (average over 500 episodes, grey) and overall (average over 1000 episodes, blue). Exploration was set to 0% in order to derive the plot.

1 and 2). However, if the training agent is not able to achieve an improvement over several super-iterations (like in the right half of Figure 1) we assume convergence. This assumption can be tested in the herein considered scenario as we know that the optimal policy for player B always wins independently of the policy of player A. Thus, if this (optimal) policy is learned, both agents should be able to win each game in which they are assigned to stance B. This is clearly the case for the right half of the above figure and similar reward curves could be derived for all 10 random argument structures and both algorithms.

As a baseline test, the trained agents played 10 games as player B against an agent based on probabilistic rules possessing the following properties: It surrenders only if no other option is left, prefers *argue* over *why* moves and chooses between equally preferred actions randomly. Thus, the agent is semi-optimal in the sense that it does not explicitly take advantage of the argument structure but follows a generally optimal strategy. As for each argument structure the trained agent wins 10 out of 10 games once the above discussed convergence is observed, we conclude that for this proof of principle case, the optimal policy was found.

6. Conclusion

We have shown how Argument Games adhering to a general structure can be reformulated as Markov games and be addressed as a multi-agent RL task. We have tested the approach on an implementation based on a specific Argument Game and random argument structures and showed that the optimal policy can be found in each considered case. Future work will focus on different aspects. In order to deal with more realistic and thus more complex scenarios, we plan to modify the Argument Game (especially the winning criterion) in order to learn more human-like strategies and also to make the game more complex. Lastly, we aim at an interaction of this system with human users that includes a competition in playing the (modified) Argument Game.

Acknowledgements: This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ”How to Win Arguments - Empowering Virtual Agents

to Improve their Persuasiveness", Grant Number 376696351, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

[1] Tangming Yuan, David Moore, Chris Reed, Andrew Ravenscroft, and Nicolas Maudet. Informal logic dialogue games in human–computer dialogue. *The Knowledge Engineering Review*, 26(2):159–174, 2011.

[2] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[3] Emmanuel Hadoux, Aurélie Beynier, Nicolas Maudet, Paul Weng, and Anthony Hunter. Optimization of probabilistic argumentation with markov decision models. In *IJCAI*, pages 2004–2010, 2015.

[4] Ariel Rosenfeld and Sarit Kraus. Strategical argumentative agent for human persuasion. In *ECAI*, pages 320–328, 2016.

[5] Chris Reed and Timothy Norman. *Argumentation machines: New frontiers in argument and computation*, volume 9. Springer Science & Business Media, 2003.

[6] Henry Prakken. Coherence and flexibility in dialogue games for argumentation. *Journal of logic and computation*, 15(6):1009–1040, 2005.

[7] Merwan Barlier, Julien Perolat, Romain Laroche, and Olivier Pietquin. Human-machine dialogue as a stochastic game. In *16th Annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL 2015)*, 2015.

[8] Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

[9] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, volume 157, pages 157–163, 1994.

[10] Michael Bowling and Manuela Veloso. Rational and convergent learning in stochastic games. In *International joint conference on artificial intelligence*, volume 17, pages 1021–1026. LAWRENCE ERLBAUM ASSOCIATES LTD, 2001.

[11] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[12] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[13] Henry Prakken and Gerard Vreeswijk. Logics for defeasible argumentation. In *Handbook of philosophical logic*, pages 219–318. Springer, 2001.

[14] Henry Prakken. Formal systems for persuasion dialogue. *The knowledge engineering review*, 21(2):163–188, 2006.

[15] Henry Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In *JELIA*, pages 224–238. Springer, 2000.

[16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.