Constraint Propagation and Reverse Multi-Agent Learning

Czechowski, A.T.

Publication date
2021

Document Version
Accepted author manuscript

Citation (APA)
Czechowski, A. T. (2021). *Constraint Propagation and Reverse Multi-Agent Learning*. Paper presented at COMARL AAAI 2021, Palo Alto, California, United States.

Constraint Propagation and Reverse Multi-Agent Learning

Aleksander Czechowski[1]
[1] Delft University of Technology
a.t.czechowski@tudelft.nl

Abstract

The development of multi-agent reinforcement learning has been largely driven by the question of how to design learning algorithms to reach some particular notion of optimality of strategies, e.g. Nash equilibria. The set of optimal strategies is not known before the execution of the learning algorithm, however we can often immediately identify a set of clearly undesirable outcomes. Therefore, we propose to consider a dual problem: given a collection of agent algorithms and a collection of unwanted strategy profiles, can one identify a set of starting strategies that invariably lead there? This leads us to study the algorithmic problem of backpropagation of constraints defining the forbidden region by learning dynamics, through the lens of set-valued maps and interval arithmetics.

Introduction

In the near future we can expect that autonomous agents will operate independently in shared environments, learn and adapt in order to achieve their individual goals. Most notorious obstacles that prevent us from reaching this vision, and deploying existing RL algorithms *en masse* are the issues of safety and reliability (Dulac-Arnold, Mankowitz, and Hester 2019). Real world systems come with constraints, which represent e.g. resource scarcity, safety limitations etc. and need to be taken into account during the learning process. As an example, let us consider the scenario of multiple self-driving cars on a shared road. The vehicles adapt their strategies based on the individual goals, like destination, desired speed etc. At the same time, certain combinations of vehicle strategies (*strategy profiles*) can be undesirable; for instance one would not want to allow strategy profiles that let two vehicles occupy the same part of the road space. In a more general setting, one can imagine that the forbidden region can be e.g. a degenerate Nash equilibrium, which one would like to avoid during the learning process; or a region in the strategy space where the rewards fall below some accepted level.

Existing works on this problem focus on the prescriptive approach, where the algorithms are designed to satisfy certain sets of constraints. The problem was considered in many papers in the single-agent case, see (Altman 1999) (Achiam et al. 2017) and recently also gained traction in the multi-agent case (Diddigi et al. 2019). Within this report we would like to explore a complementary, descriptive approach: given the dynamics of a learning algorithm and the forbidden region within the space of joint strategies, can we identify the set of strategies that are *a priori* permissible, however eventually lead to degradation into the forbidden region? There is a strong real-life motivation for answering such question. First of all, by finding all strategy profiles that degenerate to unfavourable outcomes, one can make a better informed decision on how to initialize the strategies of the agents that one controls – even with uncertainty about the exact strategies which other agents are playing. Secondly, for real-life applications it is of utmost importance to recognize the early signs that can lead to system degeneration. Then, enough time can be given for human intervention, or application of fallback algorithms.

To identify the preimage of a strategy profile under learning dynamics, one needs to perform operations which invert the learning process; in other words, starting from a particular unwanted strategy profile, obtain trajectories of strategy profiles that evolve to it. This leads us to our first research question: which algorithms for multi-agent learning can be reversed, and how? In such generality, it is difficult to answer it directly. However, for simple, deterministic models of learning behaviour known from repeated normal form games, the answer is easy. For instance, when the algorithm is defined as a solution to an initial value problem of a differential equation, it is enough to consider an initial value problem of a reversed differential equation; and when the algorithm can be represented as a deterministic mapping from a set of strategies into itself, one can try to invert it.

The other challenge is related to the cardinality of the forbidden region. As the name suggests, this set can consist of a collection of strategy profiles, which can possibly be infinite, or even dense. Therefore, the reverse learning algorithm needs to be applicable to sets, rather than points in the strategy space. This leads us to the second theme of the proposed research: how can (reverse) learning algorithms be used to propagate infinite collections of strategy profiles? We relate the problem to the field of

constraint propagation (Rossi, Van Beek, and Walsh 2006; Davis 1987).

Given the above difficulties, the problem of computing exact preimages of strategy profile sets under the learning dynamics appears intractable. For applications related to safety it is however enough to identify *outer approximations* of such sets, i.e. regions that contain the dynamical preimages. An obvious outer approximation is the set of all strategies, and the challenge we are faced with is to find methods of generating a possibly tight approximation given a certain computational budget.

Reverse Constraint Propagation for Normal Form Games

In this section we illustrate the ideas of reverse constraint propagation for learning in the simplest scenario of deterministic, repeated normal form games. A normal form game is a 3-tuple $< K, \prod_{k=1}^{K} S^k, \prod_{k=1}^{K} M^k >$ where K denotes the number of players, each set S^k is a discrete set of strategies for player k, and each map $M^k : \prod_{l=1}^{K} S^l \to \mathbb{R}$ is a payoff tensor for player k – a multidimensional array which represent the payoff player k is receiving when players use a given combination of strategies. The strategy space can be extended to the continuous space of *mixed strategies* defined as a products of simplices $X = \prod_{k=1}^{K} \Delta(S^k)$. Elements of X represent the probabilities with which each player plays their respective strategies. The extended payoff for a given mixed strategy $x = (x_1, \ldots, x_k) \in X$ is defined as the expectation $u_k(x) = \mathbb{E}_{s:s_i \sim x_i}(M^k(s))$.

Following e.g. (Piliouras et al. 2014), we model a learning process for the players as a flow $\Phi : X \times J \to X$, where J is the time set (e.g. $\mathbb{R}$, $\mathbb{Z}$). The flow Φ can be generated by iterations of the learning map $F : X \to X$ (e.g. multiplicative weight update (Arora, Hazan, and Kale 2012)), or by the solution operator of the differential equation $\dot{x} = f(x)$ (e.g. infinitesimal gradient ascent and its variations (Singh, Kearns, and Mansour 2000; Bowling and Veloso 2002; Zinkevich 2003), or replicator dynamics (Zeeman 1980)). The exact formulas of f or F are typically defined as functions of mixed strategy configurations x_k and their payoffs $u_k(x)$. For a compact subset of the strategy space $A \subset X$ representing the forbidden region, our research problem can be stated as one of identifying the *basin of attraction* of A:

$$B(A) = \{x \in X : \overline{\Phi(x, J \cap [0, \infty))} \cap A \neq \emptyset\}, \quad (1)$$

where by $\bar{Z}$ we denote the topological closure of a set Z. In other words, we want to find all mixed strategies, which eventually arrive at, or accumulate in A under the learning dynamics. By reversing the dynamics of Φ, an outer approximation of the set $B(A)$ can be given by

$$B(A) \subset \Phi(W, J \cap (-\infty, 0])), \quad (2)$$

where W is a small neighborhood of A which needs to be propagated by Φ backwards in time.

Interval arithmetics

The question of algorithmic propagation of sets by a time step maps of a flow has been thoroughly addressed by the dynamical systems community with the use of *interval arithmetics* (Moore 1966). The basic idea of these methods is to enclose possibly infinite subsets of $\mathbb{R}^n$ as products of intervals and their unions, and perform computations only on their end points. For instance, one can define the elementary floating point operations of addition, subtraction, multiplication and division on intervals, such that for any operator $\cdot \in \{+, -, *, \}$ we have the following equalities

$$[x_1, y_1] \cdot [x_2, y_2] = \{x \cdot y,\ x \in [x_1, y_1], y \in [x_2, y_2]\} \quad (3)$$

and the right hand-side of (3) is given by interval. Interval arithmetic operations can be extended in a natural way to multidimensional intervals, and used to evaluate formulas which can be expressed as compositions of the elementary operations, such as polynomials, (c.f. (Tucker 2011)).

Given a map $F : \mathbb{R}^n \to \mathbb{R}^m$ composed of elementary operations, one can leverage their interval extensions to produce an *interval enclosure* of F, denoted by $[F]$, which is defined as a map from interval sets of $\mathbb{R}^n$ to interval sets of $\mathbb{R}^m$ satisfying

$$F\left(\prod_i [x_i, y_i]\right) \subset [F]\left(\prod_i [x_i, y_i]\right). \quad (4)$$

In the dynamical context of flows, the map F is typically given by the flow time step operator $\Phi(\cdot, \pm h)$, and its interval enclosure $[\Phi(\cdot, \pm h)]$ can be used to compute outer approximations of $\Phi(W, \pm h)$, where W is an interval set, in a finite number of computations. Interval methods have been leveraged with much success in the field of dynamical systems, e.g. by aiding the solution of Smale's 14th problem (Tucker 2002). Several solvers which compute interval enclosures of maps and differential equations are available off-the-shelf (Dellnitz, Froyland, and Junge 2001; Wilczak and Zgliczyński 2012).

An example: the Stag Hunt game

Consider the well known Stag Hunt game, consisting of two players, each of them with two strategies: Stag and Hare. Their payoff profiles are given in Table 1.

		Player 2	
		$Stag$	$Hare$
Player 1	$Stag$	$(5, 5)$	$(0, 4)$
	$Hare$	$(4, 0)$	$(2, 2)$

Table 1: Payoffs of the Stag Hunt game.

The replicator dynamics of the game are given by the equations

$$\dot{x} = x((M\mathbf{y})_1 - \mathbf{x}^T M \mathbf{y}) \quad \dot{y} = y((\mathbf{x}^T N)_1 - \mathbf{x}^T N \mathbf{y}) \quad (5)$$

defined on a unit square domain $(x, y) \in [0, 1]^2$ with $\mathbf{x} = [x, 1 - x]^T$, $\mathbf{y} = [y, 1 - y]^T$, $M = \begin{bmatrix} 5 & 4 \\ 0 & 2 \end{bmatrix}$ and $N = M^T$. The variable x represents the proportion of time the Stag strategy is played by Player 1, and the variable y represents the proportion of time the Stag strategy is

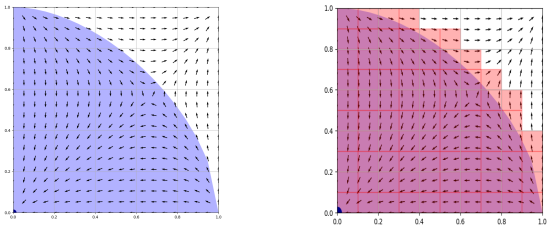

Figure 1: The vector field of the Stag Hunt game; left: the stable manifold of the (Hare,Hare) equilibrium; right: the outer approximation of the stable manifold by interval sets.

played by Player 2. The two pure Nash equilibria of the system are $(x_0, y_0) = (0, 0)$ where both players always play Hare, and $(x_1, y_1) = (1, 1)$, where both players always play Stag. Now assume, that due to an unfavourable payoff profile, we would like to exclude all strategy profiles that converge to (x_0, y_0). This means that we have to compute the basin of attraction $B((x_0, y_0))$. In this particular case the set is also known as the stable manifold of (x_0, y_0) and was computed analytically in (Panageas and Piliouras 2016). Interval aritmetics allow us for a more general, algorithmic treatment: the outer approximation of the stable manifold can be computed by propagation of small interval sets containing the equilibrium (Dellnitz and Hohmann 1997; Dellnitz, Froyland, and Junge 2001) under the reversed vector field. This yields an outer approximation which consists of a union of interval sets. We depict the shape of the basin of attraction and its interval enclosure in Figure 1.

Challenges

The question of safety in multi-agent learning leads us to the practical problem of computing preimages of forbidden regions in spaces of strategy profiles. Inspired by the possibilities offered by set valued propagators in dynamical systems, we pose several questions, which outline a plan for further research in this direction:

- Can interval arithmetics be used to produce general algorithms for computation of basins of attraction under learning dynamics in normal form games?
- How to define reverse learning in stochastic and sequential environments, so it is applicable to e.g. Markov games (Littman 1994)?
- Can the state-of-the-art multi-agent learning methods (e.g. based on policy gradients) be reversed, and can one compute their interval enclosures to backpropagate constraints?
- How can the ideas of abstraction in reinforcement learning (Sutton, Precup, and Singh 1999) be used to enhance the scalability of constraint propagation by reducing the problem dimensions?
- What are the domains of application of multi-agent learning, where identification of preimages of forbidden regions can be of most value?

We hope that by performing further investigation on the above topics, the methods of interval-based constraint propagation can be developed to become a practical tool for addressing safety questions in multi-agent learning.

Acknowledgments

This project had received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE). I would like to thank prof. Frans Oliehoek for his helpful comments on the initial version of the manuscript.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 22–31. JMLR. org.

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Arora, S.; Hazan, E.; and Kale, S. 2012. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing* 8(1):121–164.

Bowling, M., and Veloso, M. 2002. Multiagent learning using a variable learning rate. *Artificial Intelligence* 136(2):215–250.

Davis, E. 1987. Constraint propagation with interval labels. *Artificial intelligence* 32(3):281–331.

Dellnitz, M., and Hohmann, A. 1997. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numerische Mathematik* 75(3):293–317.

Dellnitz, M.; Froyland, G.; and Junge, O. 2001. The algorithms behind GAIO—set oriented numerical methods for dynamical systems. In *Ergodic theory, analysis, and efficient simulation of dynamical systems*, 145–174. Springer.

Diddigi, R. B.; Reddy, D.; KJ, P.; and Bhatnagar, S. 2019. Actor-critic algorithms for constrained multi-agent reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 1931–1933. International Foundation for Autonomous Agents and Multiagent Systems.

Dulac-Arnold, G.; Mankowitz, D.; and Hester, T. 2019. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*. Elsevier. 157–163.

Moore, R. E. 1966. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, NJ.

Panageas, I., and Piliouras, G. 2016. Average case performance of replicator dynamics in potential games via computing regions of attraction. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 703–720. ACM.

Piliouras, G.; Nieto-Granda, C.; Christensen, H. I.; and Shamma, J. S. 2014. Persistent patterns: Multi-agent learning beyond equilibrium and utility. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 181–188. International Foundation for Autonomous Agents and Multiagent Systems.

Rossi, F.; Van Beek, P.; and Walsh, T. 2006. *Handbook of constraint programming*. Elsevier.

Singh, S.; Kearns, M.; and Mansour, Y. 2000. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, 541–548. Morgan Kaufmann Publishers Inc.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1-2):181–211.

Tucker, W. 2002. A rigorous ODE solver and Smale's 14th problem. *Foundations of Computational Mathematics* 2(1):53–117.

Tucker, W. 2011. *Validated numerics: a short introduction to rigorous computations*. Princeton University Press.

Wilczak, D., and Zgliczyński, P. 2012. Cr-Lohner algorithm. *Schedae Informaticae* 2011(Volume 20).

Zeeman, E. C. 1980. Population dynamics from game theory. In *Global theory of dynamical systems*. Springer. 471–497.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 928–936.