Safety Guarantees in Multi-agent Learning via Trapping Regions

Extended Abstract

Aleksander Czechowski
Delft University of Technology
Delft, The Netherlands
acze.uni@gmail.com

Frans A. Oliehoek
Delft University of Technology
Delft, The Netherlands
f.a.oliehoek@tudelft.nl

ABSTRACT

One of the main challenges of multi-agent learning lies in establishing convergence of the algorithms, as, in general, a collection of individual, self-serving agents is not guaranteed to converge with their joint policy, when learning concurrently. This is in stark contrast to most single-agent environments, and sets a prohibitive barrier for deployment in practical applications, as it induces uncertainty in long term behavior of the system. In this work, we propose to apply the concept of trapping regions, known from qualitative theory of dynamical systems, to create safety sets in the joint strategy space for decentralized learning. Upon verification of the direction of learning dynamics, the resulting trajectories are guaranteed not to escape such sets, during the learning process. As a result, it is ensured, that despite the uncertainty over convergence of the applied algorithms, learning will never form hazardous joint strategy combinations.

KEYWORDS

multi-agent learning; safety sets; learning dynamics

ACM Reference Format:
Aleksander Czechowski and Frans A. Oliehoek. 2023. Safety Guarantees in Multi-agent Learning via Trapping Regions: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

In the recent years, enormous progress has been made for single agent planning and learning algorithms, with agents matching or exceeding human performance in various tasks and games [5, 7]. The vast success of single agent learning can be partially explained by robustness and strong convergence properties of the underlying algorithms in their basic form, such as Q-learning [9] or policy gradients [8]. Despite wide interest, the same cannot be however said for multi-agent learning. Even most basic models, e.g. replicator learning for normal form games, exhibit nonconvergence, cyclic or even chaotic behavior [6]. Even worse, it has been shown that in decoupled learning systems, there can be no learning rule that guarantees convergence to a Nash equilibrium [3]. These nonconvergent examples have also been found in more practical learning problems, such as Generative Adversarial Networks [4]. The lack of convergence guarantees in such general settings forms a major obstacle for introduction of online learning systems in practical applications, as it introduces a lot of uncertainty over what will be the state of the system, if learning is left unsupervised. Can we nevertheless still establish a type of safety certificates, that would allow us to conclude that simultaneous learning will not spin out of control?

In this paper, we suggest a novel approach to address issue. We start from the realization, that convergence is often not absolutely necessary for reliability. From systems designers perspective, it is often enough to know that learning has *rough* stability guarantees – that is, that agents will not leave a predetermined region of the strategy space during learning. We propose a method of *a priori* verifying these constraints, by establishing *trapping regions*; regions of strategy space, which learning trajectories will never escape. The idea behind this concept is simple: a candidate set for a trapping region is formed by the constraints imposed by practical, problem-dependent safety considerations. By verifying whether such set is forward-invariant for the joint learning operator, we obtain a yes–or–no answer on whether it is safe to allow concurrent multi-agent learning, without breaking these constraints.

This manuscript is an extended abstract. The complete version of the paper can be found online [2].

2 PRELIMINARIES

We consider decentralized learning schemes for groups of n agents that can be represented compactly by discrete adaptive dynamics of the form:

$$x_{t+1} := x_t + \gamma F(x_t) \tag{1}$$

with $F = [F_1, \ldots, F_n]^T$ and $x = [x^1, \ldots, x^n]^T$, where $x_i \in X_i \subset \mathbb{R}^{k_i}$ represents a point in the strategy space of a given agent i (e.g. weights in a neural network or ratios of playing a mixed strategy), and the parameter $\gamma \in \mathbb{R}^+$ denotes the adaptation rate. Throughout this paper, we assume that the learning operators are continuous, and we denote by $N = \sum_i k_i$ the dimensionality of the joint learning space. The maps $F_i : X_i \to \mathbb{R}^{k_i}$ represent the *learning operators*, i.e. the outputs of the algorithms of each agent based on the inputs. Joint strategy sequences $\{x_t\}_t$ which satisfy (1) will be referred to as the *learning trajectories*.

An *equilibrium* for the system (1) is a point in the joint strategy space $x_* \in \mathbb{R}^N$ such that $F(x_*) = 0$. In general multi-agent setting, learning schemes given by systems of form (1) do not necessarily converge to equilibria, and can have complicated, even chaotic dynamics, and might not converge to equilibria at all, as for instance in relatively simple two-player games [6].

3 TRAPPING REGIONS

In what follows, we will denote by $\operatorname{int} X$ and ∂X respectively the topological interior and boundary of a set X, and by $\operatorname{diam}(X)$ the diameter of a set X.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

Algorithm 1 Trapping region verification via binary space partitioning.

```
Inputs: Learning dynamics $F$,
$\mathbf{T} = [x_-^{11}, x_+^{11}] \times \cdots \times [x_-^{nk_n}, x_+^{nk_n}]$ – a candidate for the trapping region,
$L$ – upper bound for Lipschitz constant of $F$ over $\mathbf{T}$.
Returns: Is $\mathbf{T}$ a trapping region?
Start:
for agent i in 1:$n$ in parallel do
  for coordinate j in 1:$k_n$ in parallel do
    for direction in {left,right} in parallel do
      if direction is left then
        SETS_TO_CHECK = $\{\mathbf{T}_l^{ij}\}$, $\delta = -1$
      else
        SETS_TO_CHECK = $\{\mathbf{T}_r^{ij}\}$, $\delta = 1$
      while SETS_TO_CHECK $\neq \emptyset$ do
        $S$ = SETS_TO_CHECK.POP()
        $C(S)$ = baricenter($S$)
        if $\delta F_{ij}(C(S)) \geq 0$ then
          return false
        else if $\delta F_{ij}(C(S)) + L\,\mathrm{diam(S)}/2 \geq 0$ then
          $S_1, S_2$=SPLIT($S$) // binary partitioning
          SETS_TO_CHECK.PUSH($S_1$, $S_2$)
return true
```

Definition 1. *c.f. [1]. Let* $\mathbf{T} \subset \mathbb{R}^N$ *be a compact subset of the joint strategy space, and let* $\gamma > 0$. *If*

$$x + \gamma F(x) \subset \operatorname{int} \mathbf{T}, \quad \forall x \in \mathbf{T}, \tag{2}$$

then we call $\mathbf{T}$ *a trapping region (for the system* (1)*, with learning rate* γ*).*

In practice, verification of condition (2) can be reduced to evaluation on the boundary of $\mathbf{T}$.

Lemma 1. *Given a compact set* $\mathbf{T}$*, if* $\gamma > 0$ *is sufficiently small, and for all* $x \in \partial\mathbf{T}$ *we have*

$$x + \gamma F(x) \in \operatorname{int} \mathbf{T}, \tag{3}$$

then $\mathbf{T}$ *is a trapping region.*

Learning trajectories starting in the trapping regions are guaranteed to never leave it; furthermore, existence of a convex trapping region guarantees the existence of a learning equilibrium within.

Theorem 1. *Let* $\mathbf{T}$ *be a trapping region. Then*

(1) *Any learning trajectory* (1) *that starts in* $\mathbf{T}$ *never leaves* $\mathbf{T}$,
(2) *If* $\mathbf{T}$ *is convex, then there exists a learning equilibrium* $x^* \in \operatorname{int} \mathbf{T}$.

4 EXAMPLE

In this Section we will provide examples of application of Algorithm 1 to a toy system with known dynamics. Two other applications, in traffic management, and in the Cournot model of economic competition, can be found in the full version of the paper.

Our system exemplifies the convergence problem in multi-agent learning, but where trapping regions can be readily constructed. Both agents use gradient descent on their respective loss functions, with a fixed step γ, which leads to following update rules

$$\begin{aligned} \psi_{t+1} &:= \psi_t - \gamma(4\psi_t^3 + \epsilon\theta_t), \\ \theta_{t+1} &:= \theta_t - \gamma(4\theta_t^3 - \epsilon\psi_t). \end{aligned} \tag{4}$$

This system in fact has the same update rules as the famously non-convergent Dirac-GAN example in [4] with Wasserstein loss function, where both the generator and the discriminator apply an L^4 regularization term weighted by factor inversely proportional to ϵ. The dynamics of (4) are surprisingly complicated. It possesses a single equilibrium $(\psi, \theta) = (0, 0)$. For joint optimization, the equilibrium is always locally unstable, and the learning trajectories starting from its near proximity diverge from it until they enter a cyclic regime. For initial conditions of larger norm, they converge towards the cyclic attractor, and never reach the equilibrium; in fact none of the other trajectories does. On the other hand, it is easy to find trapping regions. We report that by Algorithm 1 we have successfully established existence of various trapping regions for different values of ϵ:

- $\mathbf{T} = [-0.1, 0.1]^4$ and $\epsilon \in \{0.01, 0.02, 0.03, 0.04\}$;
- $\mathbf{T} = [-0.2, 0.2]^4$ and $\epsilon \in \{0.05, 0.1, 0.15\}$.

For this particular system, we can also prove the existence of an ϵ-parameterized family of trapping regions theoretically, by the following proposition:

Proposition 1. *The square given by* $[-\sqrt{\epsilon}, \sqrt{\epsilon}]^2$ *is a trapping region for step size* $\gamma > 0$ *small enough. As a consequence, trajectories never leave* $[-\sqrt{\epsilon}, \sqrt{\epsilon}]^2$*, and there is an equilibrium inside* $[-\sqrt{\epsilon}, \sqrt{\epsilon}]^2$ *(it is in fact the global Nash equilibrium* $(0, 0)$*).*

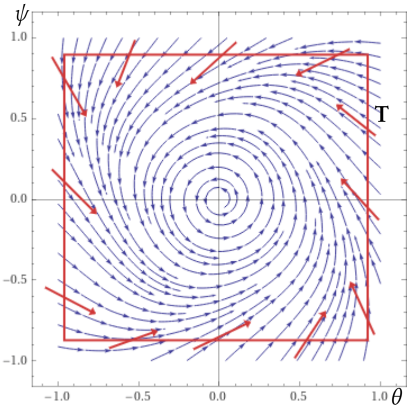

Figure 1: Trapping region in a regularized Dirac-GAN learning system.

5 CONCLUSIONS

In this paper we have applied algorithms for verification of existence of trapping regions to partially circumvent the problem of non-convergence in multi-agent learning. We have also demonstrated an application of the theory to a simple, non-convergent Generative Adversarial Network.

ACKNOWLEDGMENTS

This project was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 758824 —INFLUENCE).

REFERENCES

[1] Christian Bonatti. 2006. Generic Properties of Dynamical Systems. *Encyclopedia of Mathematical Physics* (2006), 494–502.

[2] Aleksander Czechowski and Frans A. Oliehoek. 2023. Safety Guarantees in Multi-agent Learning via Trapping Regions. https://arxiv.org/abs/2302.13844

[3] Sergiu Hart and Andreu Mas-Colell. 2003. Uncoupled Dynamics Do Not Lead to Nash Equilibrium. *American Economic Review* 93, 5 (December 2003), 1830–1836. https://www.aeaweb.org/articles?id=10.1257/000282803322655581

[4] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which training methods for GANs do actually converge?. In *ICML*. 3481–3490.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with deep reinforcement learning. *NIPS Deep Learning Workshop* (2013).

[6] Yuzuru Sato, Eizo Akiyama, and J Doyne Farmer. 2002. Chaos in learning a simple two-person game. *PNAS* 99, 7 (2002), 4748–4751.

[7] 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. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

[8] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. 1999. Policy gradient methods for reinforcement learning with function approximation.. In *NIPS*, Vol. 99. 1057–1063.

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