An Exploration Strategy Facing Non-Stationary Agents (JAAMAS Extended Abstract)

Pablo Hernandez-Leal
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
Pablo.Hernandez@cwi.nl

Yusen Zhan
Washington State University
Pullman, Washington, USA
yzhan@eecs.wsu.edu

Matthew E. Taylor
Washington State University
Pullman, Washington, USA
taylorm@eecs.wsu.edu

L. Enrique Sucar
INAOE
Puebla, México
esucar@inaoep.mx

Enrique Munoz de Cote
INAOE & PROWLER.io Ltd.
Puebla, México & Cambridge, UK
jemc@inaoep.mx

ABSTRACT

The success or failure of any learning algorithm is partially due to the exploration strategy it exerts. However, most exploration strategies assume that the environment is stationary and non-strategic. This work investigates how to design exploration strategies in non-stationary and adversarial environments. Our experimental setting uses a two agents strategic interaction scenario, where the opponent switches between different behavioral patterns. The agent's objective is to learn a model of the opponent's strategy to act optimally, despite non-determinism and stochasticity. Our contribution is twofold. First, we present *drift exploration* as a strategy for switch detection. Second, we propose a new algorithm called R-MAX# that reasons and acts in terms of two objectives: 1) to maximize utilities in the short term while learning and 2) eventually explore implicitly looking for opponent behavioral changes. We provide theoretical results showing that R-MAX# is guaranteed to detect the opponent's switch and learn a new model in terms of finite sample complexity.

Keywords

Exploration; non-stationary environments; repeated games

1. INTRODUCTION

Regardless of a task's nature and whether an agent is interacting with another agent, a human or is isolated in its environment, it is reasonable to expect some conditions will change in time. Unfortunately, a non-stationary environment invalidates assumptions behind many current learning algorithms. Examples include charging vehicles in the smart grid, poker playing, robot soccer teams and human-machine interaction scenarios. All these scenarios have one thing in common: agents must learn how their counterpart is acting and react quickly to changes in the opponent behavior.

Game theory provides the foundations for acting optimally under competitive and cooperative scenarios. However, its common assumptions that opponents are fully rational and their strategies are stationary are too restrictive to be useful in many real interactions. This work relaxes some assumptions about the environment, and focuses on the concrete setting where an opponent switches from one stationary strategy to a different stationary strategy.

We start by showing how an agent facing a stationary opponent strategy in a repeated game can treat such an opponent as a stationary (Markovian) environment, so that any RL algorithm can learn the optimal strategy against such opponents. We then show that this is not the case for non-stationary opponent strategies, and argue that classic exploration strategies (e.g. ϵ-greedy or softmax) are the cause of such failure. First, classic exploration techniques tend to decrease exploration rates over time so that the learned (optimal) policy can be applied. However, against non-stationary opponents it is well-known that exploration should not decrease so as to detect changes in the structure of the environment at all times. Second, exploring "blindly" can be a dangerous endeavour because some *exploratory actions can cause the opponent to switch strategies*, which may be harmful. Here, we use recent algorithms that perform efficient exploration [2] as stepping stone to derive a new exploration strategy against non-stationary opponent strategies.

2. DRIFT EXPLORATION

In order to motivate drift exploration, take the example depicted in Figure 1, where the learning agent faces a switching opponent in the iterated prisoner's dilemma. First, at time t_1 the opponent starts with a strategy that defects all the time, i.e., Bully. The learning agent using counts can recreate the underlying MDP that represents the opponent's strategy (i.e., the learned Bully model) by trying out all actions in all states (exploring). Second, (t_2 in the figure) the agent can solve for the optimal policy against this opponent because it has learned a correct model — the agent learns to defect, which will produce a sequence of visits to state D$_{opp}$,D$_{learn}$. Third, at time t_3, the opponent switches its selfish Bully strategy to a fair TFT strategy. But because the transition ((D$_{opp}$,D$_{learn}$), D) = D$_{opp}$,D$_{learn}$ in both MDPs, the switch in strategy (Bully $\rightarrow$ TFT) will never be visited by the learning agent.

This problem occurs because opponent strategies may share similarities in their induced MDP (specifically, between tran-

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

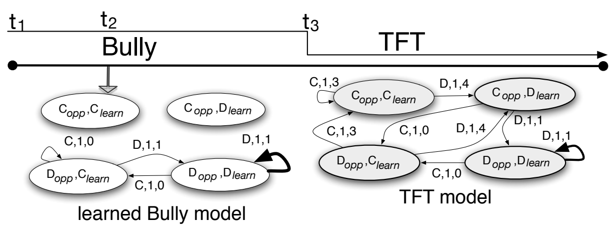

Figure 1: States are ovals — the state space is formed by the last action (C and D) of the learning agent (*learn*) and the opponent (*opp*). Each arrow has a corresponding triplet: learning agent action, transition probability and reward to the next state. An example of a learning agent against a Bully-TFT switching opponent. This example shows two MDPs: the opponent starts with Bully at time t_1 and switches to TFT at time t_3.

sition functions) when they share the same state and action spaces. This happens when the agent's optimal policy produces the same ergodic set of states, for two or more opponent strategies. It turns out this is not uncommon; for example the optimal strategy against Bully produces the sole state D_{opp},D_{learn}, however, this part of the MDP is shared with TFT. Therefore, if the agent is playing against Bully, and is stuck in that ergodic set, and a change of the opponent strategy to (say) TFT will pass unnoticed. After this shift, the unobservant learning agent will no longer be using the optimal strategy. This effect is known as *shadowing* and can only be avoided by continuously checking far visited states. The solution to this is to explore even when an optimal policy has been learned. Exploration schemes like ϵ-greedy or softmax, can be used for such purpose and will work as drift exploration with the added cost of not efficiently exploring the state space.

Against this background, our novel approach to drift exploration allows us to efficiently explores the state space, looking for changes in the transition function. The core idea of our new algorithm, R-MAX# [4], is to force the agent to revisit state-action pairs. These pairs 1) are considered to be known and 2) have not been updated in τ rounds. To encourage exploration, the algorithm resets its reward value for these pairs to *rmax* in order to promote exploration of that pair, which implicitly rechecks to determine if the opponent model has changed. We show that R-MAX# will eventually relearn a new model for the MDP after the opponent switches and compute a near-optimal policy [4].

THEOREM 1. *Let* $\tau = \frac{2m|S||A|T}{\epsilon}\log\frac{|S||A|}{\delta}$ *and M' be the new L-round MDP after the opponent switches its strategy. For any* $\delta > 0$ *and* $\epsilon > 0$, *the R-MAX# algorithm guarantees an expected return of* $U^*_{M'}(c_t) - 2\epsilon$ *within* $O(\frac{m|S|^2|A|T^3}{\epsilon^3}\log^2\frac{|S||A|}{\delta})$ *timesteps with probability greater than* $1-\delta$, *given timesteps* $t \le L$.

We tested two different domains: the iterated prisoner's dilemma and a negotiation task. Our results suggest that switch detection mechanisms [3] were not enough to deal with non-stationary opponent. Similarly, R-MAX [2] failed to update its model, and when keeping a non-decreasing learning rate and exploration WOLF-PHC [1] is capable of adapting, but it does so slowly (see Figure 2). The general approach of drift exploration by means of ϵ-greedy or softmax, solves the problem since this exploration re-visits some parts of the state space that eventually will lead to detect

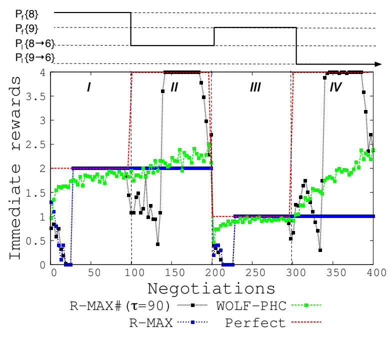

Figure 2: On top its depicted how the opponent changes among four strategies during the interaction. Immediate rewards of R-max# ($\tau = 90$), R-max and WOLF-PHC in a negotiation domain.

switches in the opponent strategy. However, the main limitation is that such algorithms are not very efficient since they explore in an undirected way. R-MAX#which implicitly handles drift exploration, is generally better equipped to handle non-stationary opponents of different sorts.

3. CONCLUSIONS

Our contribution is twofold. First, we present drift exploration as a strategy for switch detection and encode this in a new algorithm, MDP-CL(DE). This approach keeps exploring during the complete interaction and performs an undirected exploration of the state space. The second contribution is a new algorithm with implicit drift exploration, R-MAX# which makes efficient use of exploration experiences, taking into account which parts of the space state need to be re-visited, which results in rapid adaptation and efficient drift exploration, to deal with the non-stationary nature of the opponent behavior. We provided theoretical results of R-MAX# that guarantee switch detection and near-optimal expected rewards. Experiments performed in two domains showed that R-MAX# with a good parameterization converges to the optimal policy and is capable of adapting quickly to non-stationary opponent strategies either deterministic or stochastic. Its parameter, τ, shows a tradeoff: a large value enough to learn a sufficiently good model, yet small value to promote exploration for possible switches.

Acknowledgments

This research has received funding through the ERA-Net Smart Grids Plus project Grid-Friends, with support from the European Union's Horizon 2020 research and innovation programme.

REFERENCES

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

[2] R. I. Brafman and M. Tennenholtz. R-MAX a general polynomial time algorithm for near-optimal reinforcement learning. *The Journal of Machine Learning Research*, 3:213–231, 2003.

[3] P. Hernandez-Leal, E. Munoz de Cote, and L. E. Sucar. A framework for learning and planning against switching strategies in repeated games. *Connection Science*, 26(2):103–122, Mar. 2014.

[4] P. Hernandez-Leal, Y. Zhan, M. E. Taylor, L. E. Sucar, and E. Munoz de Cote. An exploration strategy for non-stationary opponents. *Autonomous Agents and Multi-Agent Systems*, Oct. 2016.