Environment Shift Games: Are Multiple Agents the Solution, and not the Problem, to Non-Stationarity?

Blue Sky Ideas Track

Alexander Mey
Delft University of Technology
a.mey@tudelft.nl

Frans A. Oliehoek
Delft University of Technology
f.a.oliehoek.tudelft.nl

ABSTRACT

Machine learning and artificial intelligence models that interact with and in an environment will unavoidably have impact on this environment and change it. This is often a problem as many methods do not anticipate such a change in the environment and thus may start acting sub-optimally. Although efforts are made to deal with this problem, we believe that a lot of potential is unused. Driven by the recent success of predictive machine learning, we believe that in many scenarios one can predict when and how a change in the environment will occur. In this paper we introduce a blueprint that intimately connects this idea to the multiagent setting, showing that the multiagent community has a pivotal role to play in addressing the challenging problem of changing environments.

KEYWORDS

Non-Stationarity; Sequential Decision Making

ACM Reference Format:
Alexander Mey and Frans A. Oliehoek. 2021. Environment Shift Games: Are Multiple Agents the Solution, and not the Problem, to Non-Stationarity?: Blue Sky Ideas Track. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 5 pages.

1 INTRODUCTION

In the past decade machine learning (ML) has made some major technological advancements and this has impact on our daily lives. This may be through the more apparent interactions with a virtual assistant or face recognition in smart devices, but it may also affect our lives in non-obvious ways. Tasks like credit score rating, judicial decisions or hiring employees are jobs that might be, and sometimes are already, done by an autonomous system [3, 23, 39].

However, many of the deployed systems are trained on a *fixed* data set, and tend to be very brittle to changes in the data distribution [24, 30, 43]. Changes in the data distribution are also called a *distribution/environment shift*, or, more general, a *non-stationary distribution/environment* [32]. For instance, the sensors of a cleaning robot may vary (due to wear) over time, and start sending slightly different signals. Similarly, an automatic traffic light controller might observe mostly light traffic, while being suddenly exposed to a traffic rush. In both cases one can imagine, that a system starts acting sub-optimally if it does not adapt. Therefore, a large number of different approaches (see Section 2) try to deal with such shifts, by detecting and reacting to the changes. However, in many cases we might be able to actually observe information that is correlated with the shifts. E.g., the sensor that wears down may do so in a certain, predictable, pattern.

For that reason we advocate that we should not merely react, but **pro-actively predict shifts**. If our decisions in turn also influence the environment, one can even imagine to **steer the environment to our benefit through those decisions**. This principle becomes in particular important regarding the following: in many cases, the decisions of an intelligent system will end up influencing the very same environment that it is trying to predict and control. For instance, imagine a company that employs a machine learning model to predict consumer demand. The model predicts that hipster consumers will like a particular fashion item, and subsequently the company saturates the market with this item. It will make initially good sales, but over time demand will wear off, *because* of the decision to saturate the market. Similarly we should expect that people adjust their behavior and/or digital profile the more that these are subject to high stake predictions by machine learning models.

As such, the more intelligent systems we employ, the more important it is to properly account for the influence of our predictions and actions. If we fail to do this, the best case is that the system starts acting sub-optimally. In the worst case, people's lives and rights are immorally and/or illegally affected [31]. Preventing this failure of technology is one of the major challenges that AI faces. In order to prevent these problems we argue that we need to consider the bigger picture of how intelligent systems interact with their environments: we want them be able to *predict* and even *steer* the way that the environment changes over time.

In this paper we bring a blueprint forward that does exactly that: predicting and steering distributional shifts. The main contribution of the proposed blueprint is to intimately connect distributional shifts to the multiagent systems (MAS) setting. The essence of the idea is to think of distributional shifts as observable entities, which we model as adversarial agents with limited power, in case we have no information about those entities. This leads to robust models, and with this blueprint any type of advancement in MAS directly helps to address the challenge of non-stationary environments. With this we believe that the MAS community should play a key role in addressing the influence of machine learning models and non-stationary environments.

2 CURRENT APPROACHES

The principle the blueprint exploits is to leverage information that allows us to *predict* and *steer* the shift of the environment. The

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

problem with this approach is, that the shift in the environment does not necessarily follow a Markovian behavior, meaning that we might have to remember a large portion of the past to predict the shift. If, for example, we want to predict how an advertising policy influences consumer demand, we would expect that not only the most recent advertisement plays a role.

Current approaches avoid this problem in different ways. Active approaches [8, 13, 15, 37, 40] follow a detect and adapt procedure, meaning that they are actively searching for changes and then try to adapt to it. Other methods follow a passive approach, as for example forgetting old data after a while [9, 35]. Both lines of thought have the problem that they can handle changes only after the fact, while on top of that we need a large sample size to detect and/or adapt to shifts. In comparison, our proposal is *pro-active*. We propose to predict shifts and in particular estimate the influence of our own action on the environment. The closest approach to actually predicting the shift and acting upon that is the hidden-mode Markov decision process [11, 19]. That model assumes that the environment acts in discrete *modes*, as for example a peak-hour and non-peak-hour mode. It makes, however, the critical assumption that the mode is an exogenous variable [10]: i.e., the mode can influence the system under concern, but is not influenced by it. Thus it does not allow us to model the influence of the ML system on the environment.

To move from detecting to predicting changes was also recently proposed in the context of Bayesian change point detection [2], motivating us further that predicting changes is a viable solution.

Finally, environment shifts are one of the main concerns for MAS themselves, as different agents will more often than not introduce a shift [21]. In relation to that, our proposal is not a method to solve the multiagent problem, but rather cast any situation with a non-stationary environment as a multiagent system, and then exploit the knowledge and tools of the MAS community.

3 THE BLUEPRINT

Before we formalize the blueprint we go through a motivating example, from which our framework will be derived. Imagine we are a big fashion company and have to think about our next product line. To take an optimal decision, we ideally would be able to predict future *customer demand*. Our own state and action space might be arbitrarily big, but there will be only a few *factors* in that state and action space that will influence the consumer demand, see Figure 1 for a schematic depiction. Furthermore one can assume that a change in the demand of the consumer does not affect all of our factors. This scenario motivates the following blueprint:

(1) Introduce a shift variable E for any source of non-stationarity.
(2) Identify what relevant factors *are* influenced by E.
(3) Identify what relevant factors *do* influence E.
(4) Predict E given the relevant factors.
(5) Leverage this information to take optimal decisions.

3.1 Formalization Through fPOSG

To formalize the blueprint we use a specific framework that can capture our idea in a general manner, the factored partially observable stochastic games (fPOSG) [5, 20].

Definition 3.1. A fPOSG is a tuple $\langle \mathcal{S}, b^0, \{A_i\}, \{O_i\}, \mathcal{P}, \{R_i\} \rangle$, where $i \in \{1, \cdots, n\}$ indicates the number of agents present.

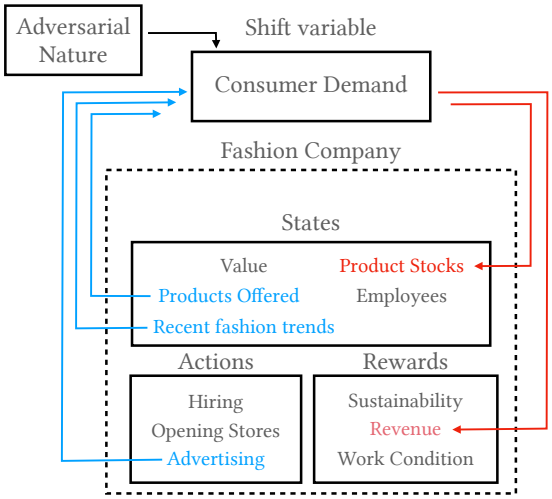

Figure 1: Depiction of the fashion example. Blue factors are the ones influencing the consumer demand (shift predictive variables), while red factors are influenced *by* the demand (shift destination variables).

- $\mathcal{S}$ is the set of environment states and they allow for a factorization as $\mathcal{S} = \prod_{j=1}^{k} S_i$.
- b^0 is an initial distribution over the states from $\mathcal{S}$.
- A_i is the action space of agent i. We define $\mathcal{A} := \prod_{i=1}^{n} A_i$ as the joint action space.
- O_i is the set of possible observations for agent i. We define $O = \prod_{i=1}^{n} O_i$ as the joint observation space.
- $\mathcal{P}$ denotes a set of transition probabilities. In particular, given a joint action $a \in \mathcal{A}$, states $s, s' \in \mathcal{S}$ and a joint observation $o \in O$, $\mathcal{P}(s', o \mid a, s)$ is the probability of receiving observations o and transition to state s', given that the agents took the joint action a while being in state s.
- $R_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function for agent i.

With the previous definitions in play we may introduce the framework for the blueprint, the *environment shift game*. The simple, yet powerful, idea is to add a distinguished set of variables to a fPOSG, which will allow us to take shifts into account and plan our own actions accordingly.

Definition 3.2. An *environment shift game* (ESG) is a fPOSG, with a state space factorization $\mathcal{S}' = \mathcal{S} \times E$, where E is a distinguished set of state factors E, the *shift variables*, and two agents: the decision maker, and an adversarial nature, who can influence the shift variables. The latter's reward function is the additive inverse of the decision maker such that the game is zero-sum.

The idea behind this definition is that, to the extent possible, we model the change in distribution. However, to deal with the limitations of our knowledge about the evolution of the shift, we apply worst-case reasoning (by assuming an adversarial nature) to derive robust plans [14, 18, 25, 38]: essentially the ('optimal') solution of a zero-sum game yields a minimax policy for the decision maker that provides security level payoff [36, 41]. Moreover, such a policy would optimally use any local information ('shift prediction

variables' below) to anticipate the impact of the shifts. Of course, in case we are be able to fully specify the probabilistic model of the shift variables, no adversarial agent is needed and the model reduces to a (single-agent) POMDP [28, 42], which is just a special case of ESG.[1] Also, the definition above can be further generalized when useful: the decision maker could actually comprise a team of decision makers, and in some cases it may be reasonable to assume that the incentives are not strictly competitive.

Going back to the previous fashion example, we make the following connections and additional definitions. Figure 1 shows the ESG structure, with the given states, actions, rewards and the shift variable, given by the consumer demand. The decision maker takes actions within the company, while the adversarial nature models the unknowable aspects of consumer demand. Importantly, we may limit the capabilities of the adversary: e.g., it may perhaps decrease demand for some products, but not for all. This way we can **encode domain knowledge and assumptions** in the model.[2] Note that the states (and similarly the actions and rewards) have different factors, the product stocks, the value etc. In Figure 1 we identify in blue the factors that *influence* the shift, and in red the factors that are *influenced by* the shift. Factors that influence the shift variable and factors that are influenced by it will be respectively called *shift predictive variables* and *shift destination variables*.

Note that the shift variables E may correspond to concrete concepts like 'demand' or 'rush hour', but they can also be abstract without a clear meaning: in that case E receives its meaning implicitly by appropriately influencing the shift destination variables (i.e., being a parent of them in a dynamic Bayesian network representation [5], cf. Figure 2 and 3). Also, we do not exactly need to know why the shift happens. Precisely when we do not fully understand how E evolves, we can avoid specifying exact transition probabilities and instead specify intervals for the adversary of what are deemed possible probabilities.

This approach is much more powerful than just an ad hoc adaptation to a shift. If a sensor wears down, and we follow a detect and adapt procedure, we constantly need to monitor and it will always take time and samples to detect the shift. The sequential reasoning of the ESG agents, however, will consider the relationship between time and wear, and thus the decision maker can anticipate the wear even before it happens. The optimal solution of an ESG is thus one that can **pro-actively predict shifts before they happen and adapt to them**, and **steer the environment to our benefit** .

3.2 Modeling other Settings as an ESG

Many settings with a non-stationary component may actually be modeled as an ESG. We illustrate this with two examples: the covariate shift and the online learning setting.

Covariate Shift. The term covariate shift is used for shifts in supervised learning problems, where we use a prediction rule $f(X) = \hat{Y}$ [32]. Specifically, covariate shift assumes that the distribution of the covariate $P(X)$ changes, while $P(Y \mid X)$ remains stationary, where Y is a response variable.

[1] We still refer to these as ESGs, since they still model the shift as an explicit entity.
[2] In many settings it is known that we cannot efficiently learn without any assumptions [1, 45], and only restrictions on the adversary make that possible [9, 35].

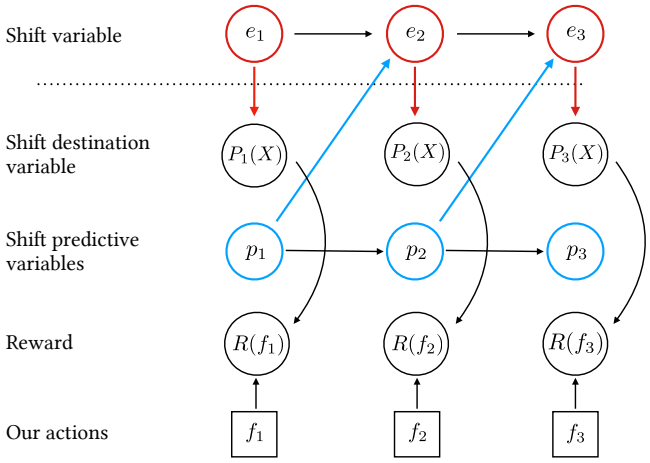

Figure 2: Covariate shift in the activity prediction example. The shift variable are the seasons, the shift predictive variable is the time, and our action is the model we want to use.

For example, think about the following setting: during the year we feed recent pictures of activities from the same location to a machine learning model f, and it is supposed to predict the type of activity seen. Throughout the seasons the performance $R(f)$ of our model is affected due to seasonality. During winter, it frequently predicts images with a lot of white as skiing, and this leads to good results. However, moving to spring and summer this bias does not work anymore, since lot of white in the picture refers rather to indoor sports. In short, the biases in the dataset are not stationary throughout the seasons.

This setting can easily be captured as an ESG, as illustrated in Figure 2. The seasonality could be encoded in the p_t variables in the ESG, and the optimal solution would essentially encode when we need to retrain our classifier. We could even expand the model by explicitly incorporating the costs involved with retraining.

Online Learning. Online learning is a sequential decision making task, where in each round t we have to choose an action a_t and another (possibly) adversarial agent choses an action n_t. Based on those two actions we receive and observe a reward (or loss) $l(a_t, n_t)$. Formulating this in our framework, we generalize the setting a bit. Our loss will depend on the shift variable e_t and the action of the adversary can merely influence e_t with its own actions n_t, see Figure 3 for a graphical depiction. This allows us to model all types of strength of the adversary, it might be able to directly chose e_t, or, for example, may only be able to only change e_t slightly. As a motivating example we consider investing in the stock market. In each time step our action is to distribute our wealth on the possible assets, and the return is the gain in wealth. The return of each asset underlies a constant shift, which is captured in the ESG.

3.3 Stationarity of the ESG

A very important question the reader may ask now: does an ESG solve the non-stationarity problem? The answer is: sometimes. It is actually clear that it cannot always solve the problem. If any part of the environment may change without regularity, there is no hope

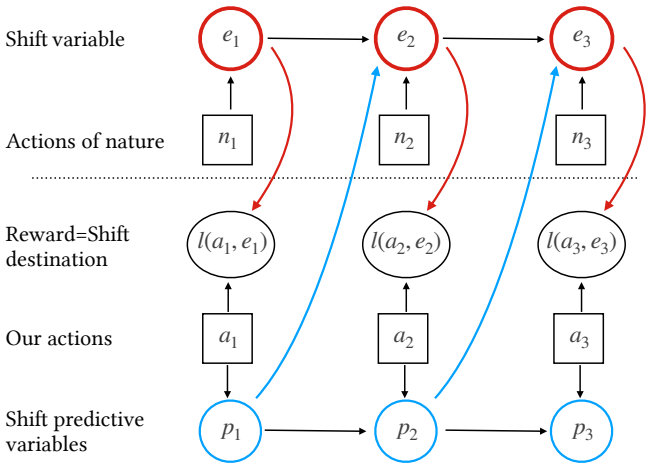

Figure 3: Online learning in the stock market. Most importantly, our own actions may influence the shift, and this is modeled by the arrow from a_t to p_t. The shift predictive variable p_t may contain more information, i.e. everything we consider relevant for prediction.

that we can account for it pro-actively. We may still adapt to it after the fact, but this is then rather in the spirit of the active or passive methods discussed in Section 2. Imagine, however, a traffic scenario that has truly only two types of transitions; one for peak-hour $\mathcal{P}_1(s' \mid s)$ and one for non-peak-hour $\mathcal{P}_2(s' \mid s)$. Without further knowledge, the switching of the environment between $\mathcal{P}_1$ and $\mathcal{P}_2$ is outside of our control. If we, however, introduce the shift variable e, which corresponds to the time of the day, we can turn the transitions into one stationary process $\mathcal{P}(s' \mid s, e)$.

But even if the ESG does not manage to turn a non-stationary problem into a stationary one, the approach is still valid and useful. If, in the traffic example, the transition functions also depend on a factor that we do not account for, say which day of the year it is, we still have a more benign problem if we at least take the time of the day into account. One may see the ESG as the attempt to turn a non-stationary problem, as good as possible, into a stationary one.

3.4 Solving the ESG

While the ESG is a very powerful framework, it is a non-trivial case of POSG and solving POSGs is far from easy [20]. Nevertheless, we believe that there is hope that the communities working on game theory and multiagent reinforcement learning (MARL) will further integrate (e.g., [29]) and provide the insights and tools to make ESGs practical. For instance, there has been tremendous progress in zero-sum games [4, 17, 46] like poker [6, 7]. Moreover, we may be able to exploit various forms of structure, such as common-knowledge [29], forms of observability [22], or structure of value function [44]. Similarly we may reduce the complexity of the problem with approaches like *influence-based abstraction* [33, 34]. The idea directly connects to our proposed framework, as one assumes that we have a local model of observed variables, and non-local variables that are unobserved, which correspond to the shift variable. Approximating the influence with neural networks or other types of function approximation is a promising direction to deal with scalability [12]. Additionally, we have seen tremendous empirical progress by combining deep learning with MARL [7, 16, 26, 27], which gives hope that we would be able to derive useful policies in practice.

4 SUMMARY AND CHALLENGES

In this paper we formalized a framework, called the environment-shift game, that intimately connects a large class of non-stationary problems to the MAS setting. We argued that this offers a number of opportunities: we can **pro-actively predict shifts before they happen and adapt to them**, we can **steer the environment to our benefit** and we can **encode domain knowledge and assumptions** (inductive bias) in the model. One may also wonder if a trained ESG can be useful to explain certain behavior of an AI system, as it tries to capture the hidden dynamics between the AI and its environment. There are, however, still many challenges that need addressing before we can deploy realistic ESGs:

Learning the model will be a big technical challenge. However, even if learning 'the correct' model might be impossible in the near future, learning *approximate* models may be feasible. We envision that this will be a better approach than ignoring the shift altogether. In certain scenarios one may also be able to perform a sensitivity analysis, to decide which parts of the model are crucial to model exactly, and which parts may be approximated.

Scalability of the action space may pose another technical challenge in some ESGs. In the covariate shift setting from Figure 2, for example, the action of the decision maker could be the selection of an entire classifier. Dealing with such complex action spaces might be difficult and a number of questions arise: Can we select good (candidate model) action subsets? Can we integrate supervised learning as a manner of action selection in a principled way? Can we exploit modularity of machine learning models, perhaps leading to factored action spaces? We may, for example, share parts between different models that are not affected by the distribution shift, for example certain layers of a neural network. **Considering ethical implications** in the context of ESGs is a topic that has to be addressed before deploying them. As pointed out earlier, the ESG will use the actions to steer the environment in a desired direction. This steering is only implicit, so in some cases it may be very hard to control what the long-term effect on the environment actually is. In cases where societal values, as fairness, safety or other, are part of the environment, it becomes critical to study the effect that the actions have on the environment. On the other hand, we can see the influence on the environment as a chance. If we align the societal values with the reward the system receives, an ESG should take actions that will create long-term benefits towards those values.

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).

REFERENCES

[1] Yasin Abbasi-Yadkori, Peter L. Bartlett, Varun Kanade, Yevgeny Seldin, and Csaba Szepesvári. 2013. Online Learning in Markov Decision Processes with Adversarially Chosen Transition Probability Distributions. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2508–2516.

[2] Diego Agudelo-España, Sebastián Gómez-González, Stefan Bauer, Bernhard Schölkopf, and Jan Peters. 2020. Bayesian Online Prediction of Change Points. In *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 320–329.

[3] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. *PeerJ Computer Science* 2 (2016), e93.

[4] Branislav Bosanský, Christopher Kiekintveld, Viliam Lisý, and Michal Pechoucek. 2014. An Exact Double-Oracle Algorithm for Zero-Sum Extensive-Form Games with Imperfect Information. *Journal of Artificial Intelligence Research* 51 (2014).

[5] Craig Boutilier, Thomas L. Dean, and Steve Hanks. 1999. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *Journal of Artificial Intelligence Research* 11 (1999), 1–94.

[6] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. 2015. Heads-up Limit Hold'em Poker is Solved. *Science* 347, 6218 (2015), 145–149.

[7] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for Multiplayer Poker. *Science* 365, 6456 (2019), 885–890.

[8] Giuseppe Canonaco, Marcello Restelli, and Manuel Roveri. 2020. Model-Free Non-Stationarity Detection and Adaptation in Reinforcement Learning. In *24th European Conference on Artificial Intelligence (Frontiers in Artificial Intelligence and Applications, Vol. 325)*. IOS Press, 1047–1054.

[9] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. 2020. Reinforcement Learning for Non-Stationary Markov Decision Processes: The Blessing of (More) Optimism. *CoRR* abs/2006.14389 (2020).

[10] Rohan Chitnis and Tomás Lozano-Pérez. 2019. Learning Compact Models for Planning with Exogenous Processes. In *3rd Annual Conference on Robot Learning (Proceedings of Machine Learning Research, Vol. 100)*. PMLR, 813–822.

[11] Samuel P. M. Choi, Dit-Yan Yeung, and Nevin Lianwen Zhang. 1999. An Environment Model for Nonstationary Reinforcement Learning. In *Advances in Neural Information Processing Systems 12*. The MIT Press, 987–993.

[12] Elena Congeduti, Alexander Mey, and Frans A. Oliehoek. 2020. Loss Bounds for Approximate Influence-Based Abstraction. *CoRR* abs/2011.01788 (2020).

[13] Bruno Castro da Silva, Eduardo W. Basso, Ana L. C. Bazzan, and Paulo Martins Engel. 2006. Dealing with non-stationary environments using context detection. In *Proceedings of the Twenty-Third International Conference on Machine Learning (ACM International Conference Proceeding Series, Vol. 148)*. ACM, 217–224.

[14] Karina Valdivia Delgado, Scott Sanner, and Leliane Nunes de Barros. 2011. Efficient Solutions to Factored MDPs with Imprecise Transition Probabilities. *Artificial Intelligence* 175, 9-10 (2011), 1498–1527.

[15] Gregory Ditzler, Manuel Roveri, Cesare Alippi, and Robi Polikar. 2015. Learning in Nonstationary Environments: A Survey. *IEEE Computational Intelligence Magazine* 10, 4 (2015), 12–25.

[16] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2137–2145.

[17] Andrew Gilpin and Tuomas Sandholm. 2008. Solving Two-Person Zero-Sum Repeated Games of Incomplete Information. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 2. IFAAMAS, 903–910.

[18] Robert Givan, Sonia M. Leach, and Thomas L. Dean. 2000. Bounded-Parameter Markov Decision Processes. *Artificial Intelligence* 122, 1-2 (2000), 71–109.

[19] Emmanuel Hadoux, Aurélie Beynier, and Paul Weng. 2014. Solving Hidden-Semi-Markov-Mode Markov Decision Problems. In *Scalable Uncertainty Management - 8th International Conference (Lecture Notes in Computer Science, Vol. 8720)*. Springer, 176–189.

[20] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. 2004. Dynamic Programming for Partially Observable Stochastic Games. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press / The MIT Press, 709–715.

[21] Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. 2017. A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity. *CoRR* abs/1707.09183 (2017).

[22] Karel Horák. 2019. *Scalable Algorithms for Solving Stochastic Games with Limited Partial Observability*. Ph.D. Dissertation. Czech Technical University in Prague.

[23] Cheng-Lung Huang, Mu-Chen Chen, and Chieh-Jen Wang. 2007. Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications* 33, 4 (2007), 847–856.

[24] Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. 2020. The Impact of Non-stationarity on Generalisation in Deep Reinforcement Learning. *CoRR* abs/2006.05826 (2020).

[25] Garud N Iyengar. 2005. Robust Dynamic Programming. *Mathematics of Operations Research* 30, 2 (2005), 257–280.

[26] Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. 2019. Human-level Performance in 3D Multiplayer Games with Population-Based Reinforcement Learning. 364, 6443 (2019), 859–865.

[27] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. 2020. Graph Convolutional Reinforcement Learning. In *8th International Conference on Learning Representations*. OpenReview.net.

[28] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence* 101, 1-2 (1998), 99–134.

[29] Vojtěch Kovařík, Martin Schmid, Neil Burch, Michael Bowling, and Viliam Lisý. 2019. Rethinking Formal Models of Partially Observable Multiagent Decision Making. *CoRR* abs/1906.11110 (2019).

[30] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *CoRR* abs/2005.01643 (2020).

[31] Anthony Macciola. 2019. Bad, biased, and unethical uses of AI. https://enterprisersproject.com/article/2019/8/4-unethical-uses-ai. Accessed: 2020-11-30.

[32] Jose G. Moreno-Torres, Troy Raeder, Rocío Alaíz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern Recognition* 45, 1 (2012), 521–530.

[33] Frans A. Oliehoek, Stefan J. Witwicki, and Leslie Pack Kaelbling. 2012. Influence-Based Abstraction for Multiagent Systems. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 1422–1428.

[34] Frans A. Oliehoek, Stefan J. Witwicki, and Leslie Pack Kaelbling. 2019. A Sufficient Statistic for Influence in Structured Multiagent Environments. *CoRR* abs/1907.09278 (2019).

[35] Ronald Ortner, Pratik Gajane, and Peter Auer. 2019. Variational Regret Bounds for Reinforcement Learning. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 16.

[36] Martin J. Osborne and Ariel Rubinstein. 1994. *A Course in Game Theory*. The MIT Press.

[37] Sindhu Padakandla, Prabuchandran K. J., and Shalabh Bhatnagar. 2019. Reinforcement Learning in Non-Stationary Environments. *CoRR* abs/1905.03970 (2019).

[38] Marek Petrik and Dharmashankar Subramanian. 2014. RAAM: The Benefits of Robustness in Approximating Aggregated MDPs in Reinforcement Learning. In *Advances in Neural Information Processing Systems 27*. 1979–1987.

[39] Manish Raghavan, Solon Barocas, Jon M. Kleinberg, and Karen Levy. 2020. Mitigating bias in algorithmic hiring: evaluating claims and practices. In *FAT* '20: Conference on Fairness, Accountability, and Transparency*. ACM, 469–481.

[40] Manuel Roveri. 2019. Learning Discrete-Time Markov Chains Under Concept Drift. *IEEE Transactions on Neural Networks and Learning Systems* 30, 9 (2019), 2570–2582.

[41] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multi-Agent Systems: Algorithmic, game-theoretic and logical foundations*. Cambridge University Press.

[42] Matthijs T. J. Spaan. 2012. Partially Observable Markov Decision Processes. In *Reinforcement Learning: State of the Art*. 387–414.

[43] Masashi Sugiyama and Motoaki Kawanabe. 2012. *Machine Learning in Non-Stationary Environments - Introduction to Covariate Shift Adaptation*. MIT Press.

[44] Auke J. Wiggers, Frans A. Oliehoek, and Diederik M. Roijers. 2016. Structure in the Value Function of Two-Player Zero-Sum Games of Incomplete Information. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence (ECAI)*. IOS Press, 1628–1629.

[45] Jia Yuan Yu, Shie Mannor, and Nahum Shimkin. 2009. Markov Decision Processes with Arbitrary Reward Processes. *Mathematics of Operations Research* 34, 3 (2009), 737–757.

[46] Kaiqing Zhang, Sham M. Kakade, Tamer Basar, and Lin F. Yang. 2020. Model-Based Multi-Agent RL in Zero-Sum Markov Games with Near-Optimal Sample Complexity. *CoRR* abs/2007.07461 (2020).